102 PROGRAMMI

PER

PHILIPS C7420 VIDEOPAC+

DI

J. DECONCHAT

V. C. GRANDIS

102 programmi per PHILIPS C7420 VIDEOPAC +

DELLO STESSO EDITORE

Volumi pubblicati

D. A. Lien - Dizionario del Basic
J. Boisgontier - Il Basic per tutti
A. Pinaud - CP/M passo dopo passo
D.-J. David - La scoperta del Commodore 64
D.-J. David - La pratica del Commodore 64
D.-J. David - L'assembler del Commodore 64 - 3. Linguaggio macchina e assembler del 6502
J. Deconchat - 102 programmi per Commodore 64
J. Boisgontier - Commodore 64: metodi pratici
J. Boisgontier, S. Brebion, G. Foucault - Il Commodore 64 per tutti
X. Linant de Bellefonds - La pratica dello ZX Spectrum - Vol. 1
M. Henrot - La pratica dello ZX Spectrum - Vol. 2
J.-F. Séhan - Chiavi per lo ZX Spectrum
J. Lévy - Esercizi per lo ZX Spectrum
J.-F. Séhan - Alla ribalta: lo ZX Spectrum
C. Galais - Vademecum per Applesoft
J. Boisgontier - L'Apple e i suoi files
B. De Merly - Guida per l'Apple - Vol. 1
B. De Merly - Guida per l'Apple - Vol. 2
B. De Merly - Guida per l'Apple - Vol. 3
F. Lévy - Esercizi per l'Apple II, II plus, IIe, IIc
J. Boisgontier - 36 programmi per Apple IIe, II plus, IIc
N. Bréaud-Pouliquen - La pratica dell'Apple II - 1. Periferiche e gestione dei file
J.-P. Blanger - Modelli di espressione grafica
A. Pinaud - Programmare in Forth
C. Bardon, B. De Merly - Giochi sul Philips C7420 Videopac +
J.P. Richard - La scoperta del PC 1500
J.-F. Séhan - Supergiochi per il Commodore 64. 30 programmi in Basic

Volumi di prossima pubblicazione

J. Deconchat - 102 programmi per ZX Spectrum e ZX 81
M. Henrot - J. Boisgontier - Lo ZX Spectrum per tutti
A. Pinaud - MS DOS passo dopo passo
M. Thibault - Chiavi per il MULTIPLAN
C. Nowakhwski - Programmare in C
N. Bréaud-Pouliquen - La pratica dell'Apple II - Vol. 2

VALENTINO C. GRANDIS - JACQUES DECONCHAT

102 programmi per PHILIPS C7420 VIDEOPAC +

Editsi - Editoriale per le scienze informatiche - S.r.l.
MILANO 1985

Sommario

Prefazione

Troverete in questo libro più di 100 programmi, redatti in Basic-Microsoft, per il VIDEOPAC+ home-computer; la maggior parte dei programmi saranno generalmente utilizzabili, con opportune modifiche su altri home-computer che usino un Basic-Microsoft.

I programmi proposti sono molto brevi, e l'ordine nel quale sono presentati non è certamente arbitrario; si tratta di iniziare il lettore, mediante la realizzazione e l'utilizzazione di programmi di giochi relativamente elementari, alla conoscenza e alla padronanza progressiva di quello strumento straordinario che è un home-computer.

La brevità dei programmi proposti permetterà di batterli sulla tastiera senza introdurre troppi errori. Potrete, naturalmente, utilizzarli come base per la realizzazione di programmi più impegnativi; i programmi sono abbondantemente commentati per aiutarvi a capire il loro funzionamento, e le pagine di presentazione vi permetteranno di ritrovare rapidamente le caratteristiche proprie delle istruzioni Basic che utilizzano.

I programmi utilizzano pressoché tutte le particolarità di un calcolatore come il VIDEOPAC+, le cui caratteristiche sono peraltro dettagliate sul Manuale d'Uso. In particolare abbiamo utilizzato nell'ultima sezione le istruzioni PEEK che consentono di leggere direttamente i byte delle posizioni di memoria ad accesso casuale RAM, per crearci la copia dello schermo.

Alla fine di ogni programma troverete qualche suggerimento per possibili migliorie o per modifiche che potrebbero rendere il gioco proposto più attraente e completo. Fra i libri che possono essere d'ausilio per creare i vostri giochi è stato tradotto in italiano: *"Giochi sul Philips C7420 VIDEOPAC+"* di Christophe Bardon e Benoît de Merly, Ed. EPSI, Milano, 1985.

ING. VALENTINO GRANDIS

Milano, febbraio 1985

Organizzazione dell'opera

I giochi sono classificati in ordine crescente di difficoltà seguendo 5 livelli:

— Il **LIVELLO 1** permette di studiare le istruzioni più elementari del linguaggio Basic: **LET** (opzionale qui) **PRINT, INPUT, IF...THEN, FOR ...NEXT, GOTO, GOSUB, ON GOTO** — .

— Al **LIVELLO 2** si utilizzeranno le istruzioni di confezione e di manipolazione degli elementi di un array, e della ricerca di un elemento in una stringa: **DIM, R(I), MID$, READ, DATA**.

— Al **LIVELLO 3** si utilizzerà la serie completa delle istruzioni di utilizzo di una stringa di caratteri: **MID$, RIGHT$, LEFT$**.

— Al **LIVELLO 4** sarà utilizzata un'istruzione molto particolare, che facilita la realizzazione di giochi interattivi: **KEY (0)**.

— Per finire, il **LIVELLO 5** ci farà scoprire le istruzioni-frontiera del Basic, che permettono la manipolazione diretta nelle locazioni di memoria dell'apparecchio, la definizione dei caratteri grafici speciali e l'identificazione di oggetti sullo schermo: **SETEG, POKE**.

Ogni gioco è presentato, in maniera molto sistematica, da una corta descrizione, seguita dallo studio dettagliato di ognuna delle linee importanti del programma. Seguono le indicazioni necessarie al buon funzionamento del gioco, così come alcuni elementi per delle possibili estensioni o migliorie del gioco proposto.
Il programma, funzionante sul VIDEOPAC+, è in seguito proposto; seguito da un esempio di esecuzione. I listings sono stati ottenuti direttamente dagli originali degli autori, su una stampante matriciale, il che dovrebbe garantire, contro ogni rischio di malfunzionamento, il programma che è stato usato alla stampa. Questo programma, molto elementare (non accetta alcuni caratteri), è fornito alla fine della spiegazione.

LIVELLO 1

N.	Nome del programma	Tema e particolarità
1	Il numero misterioso	Gioco di deduzione a più livelli, per un giocatore.
2	Il cubo	Gioco di disegno: prospettiva cavaliera su schermo.
3	Caldo e freddo	Logica e caso a più livelli, per un giocatore.
4	I fiammiferi	Gioco di riflessione contro l'elaboratore.
5	I fiammiferi in molti	Gioco di riflessione e di strategia, per più giocatori.
6	Calendario perpetuo	Preparazione di un calendario, mese per mese.
7	Battaglia navale	Una sola nave da trovare, ma con qualche particolarità.
8	Golf	Un percorso a 18 buche, per un solo giocatore.
9	Solfeggio	Imparate il solfeggio per mezzo del vostro elaboratore.
10	Blackjack (Gioco del 21)	Gioco d'azzardo, l'elaboratore tiene il banco.
11	Pari-dispari	Gioco d'azzardo in tre tempi con uno o più giocatori.
12	Indovina un numero	Un gioco facilissimo per uno o più giocatori.
13	Il gioco di Wytthof	Gioco di riflessione molto semplice, per due giocatori.
14	Rimbalzi	Gioco d'abilità, assai visivo.
15	Jackpot (Pentolaccia)	Gioco d'azzardo, per un giocatore.
16	Roulette	Un'elegante simulazione di un gioco assai noto.
17	Corridoi	Un gioco di logica tutto da scoprire.
18	Scacchiera	Una base grafica per le vostre creazioni.
19	Il gioco della fava	Gioco di riflessione, per due giocatori.
20	Gioco dell'oca aleatorio	Un antichissimo gioco, riveduto e corretto dall'elaboratre, per due giocatori.

Istruzioni elementari

Non tratteremo qui che le istruzioni fondamentali, o istruzioni di programmazione; molti altri comandi, o funzioni particolari, saranno utilizzati quando occorre, per esempio:

— ***RND(1) per ottenere un numero aleatorio tra 0 e 1;***
— ***INT per prendere la parte intera di un risultato;***
— ***REM per mettere un commento;***
— ***END per terminare un programma, ecc.***

Le istruzioni fondamentali della programmazione, utilizzate già al livello 1 e presenti in praticamente tutti i programmi, sono:

LET (opzionale); **PRINT**; **INPUT**; **GOTO|___|**; **IF|___|THEN|___|**; **FOR ...NEXT**; **GOSUB...RETURN**

ossia solamente sette istruzioni di base (e due istruzioni derivate) che esamineremo ora più in dettaglio.

L'ISTRUZIONE LET, O ISTRUZIONE DI ASSEGNAZIONE

È l'istruzione per eccellenza dei linguaggi di programmazione evoluti, visto che il suo ruolo essenziale è di permettere, per il programmatore, una gestione trasparente della memoria. Questo significa, in pratica, che per inserire una quantità in memoria, per esempio 12, non è necessario sapere in quale locazione di memoria la si deporrà: basta assegnarle un nome autorizzato, ed è con questo nome che la quantità sarà in seguito ritrovata e manipolata.

— **Se la quantità è numerica**, il nome sarà una semplice lettera dell'alfabeto, o un insieme di varie cifre o lettere, cominciando obbligatoriamente con una lettera.
— **Se la quantità è alfanumerica**, si applica la stessa regola per i nomi, che dovranno però essere seguiti dal simbolo $, e la variabile dovrà essere scritta tra virgolette.

Esempio: LET A1=12; la variabile numerica A1 prende il valore 12.
LET NOME$="MARTINO"; la variabile alfanumerica NOME$ prende il valore MARTINO.

Sul VIDEOPAC+, come nella maggior parte dei Basic, il LET è diventato opzionale, non essendo possibile nessuna ambiguità; scriveremo dunque: A=12; NOME$=MARTINO; ecc.

L'ISTRUZIONE PRINT, O ISTRUZIONE DI USCITA (SCHERMO)

È un'istruzione di dialogo, che permette al calcolatore di visualizzare sullo schermo un testo, il risultato di un calcolo, o una variabile. È il solo modo che possiede il personal computer per far conoscere i suoi risultati. Così PRINT A1 provoca la visualizzazione di 12 (se era stato fatto prima A1=12); PRINT A$ provoca la visualizzazione di BUONGIORNO (se era stato fatto prima A$="BUONGIORNO"). Utilizzato dopo un cancellamento schermo, il PRINT provoca la visualizzazione sulla prima linea, in alto, a sinistra.
Un certo numero di parametri possono essere utilizzati per ottenere sullo schermo una visualizzazione adeguata:

— **PRINT SPC(M)** permette d'inserire degli spazi in una linea di risultati. Il loro numero viene precisato dall'argomento.

— **PRINT TAB(N)** provoca sulla linea di stampa corrente una visualizzazione cominciante all'ennesima colonna partendo da sinistra (1< =N< =39).

Nota. La differenza fra **SPC** e **TAB** consiste nel fatto che **TAB** calcola gli spazi partendo da inizio linea, mentre **SPC** si riferisce alla posizione corrente del cursore.

— Un **; (punto e virgola)** a seguito del PRINT permette sia di concatenare un'altra stampa senza ripetere l'istruzione PRINT, sia di indicare che la prossima stampa sarà, in assenza di un'indicazione contraria, effettuata esattamente al seguito della stampa precedente.
— La **, (virgola)** provoca una tabulazione orizzontale automatica predefinita.

Durante la battuta, l'istruzione PRINT potrà essere abbreviata battendo **? (punto interrogativo)**. Durante una stampa, ogni numero è preceduto da un segno; ma il segno positivo non è visualizzato: uno spazio precederà dunque automaticamente la stampa di ogni risultato positivo.

LE ISTRUZIONI CURSOR X E CURSOR Y

Il VIDEOPAC+ dispone di istruzioni particolari, di impiego assai flessibile, che *permettono di posizionare il cursore sullo schermo in previsione di una prossima scrittura*. Si tratta delle istruzioni: CURSOR X (n) e CURSOR Y (m) che saranno generalmente utilizzate con una tabulazione orizzontale da 1 a 39 e verticale da 0 a 22. Attenzione però ad evitare CURSOR X 0 che provoca una riga in inversione sullo schermo, mentre per CURSOR Y (0, 21 e 22) per fissare lo schermo deve essere data l'istruzione PAGE che blocca lo scorrimento verso l'alto o verso il basso (SCROLL).

COLORE DELLO SCHERMO

Per definire il colore di sfondo dello schermo si usa la specifica: INIT (i) in cui l'indice (i) rappresenta il colore di fondo e deve essere un numero intero, modulo 8, ossia pari al resto della divisione per 8.
Per cancellare i caratteri basta renderli invisibili, perciò talvolta si usa questa specifica in tal senso, dando a (i) il medesimo valore dei colori di primo piano sia di testo che grafici, definiti dalle funzioni TX e GR.
DELIM è una funzione analoga che però consente di specificare contemporaneamente (ma soltanto dalla posizione del cursore sino a fine riga) sia il colore di fondo che quello di primo piano. Il suo terzo indice consente la cancellazione, l'incasellamento e la sottolineatura.

L'ISTRUZIONE INPUT, O ISTRUZIONE D'INGRESSO (TASTIERA)

Questa istruzione, che provoca l'arresto e l'attesa del calcolatore, permetterà di inserire un numero, o un testo, che sarà assegnato alla variabile attraverso l'ordine INPUT. Questo è fatto in modo conversazionale, cioè in funzione delle necessità; infatti, quando il VIDEOPAC trova l'istruzione INPUT, si ferma e vi passa la mano; si può allora inserirgli un'informazione di natura adeguata (numero o testo), e segnalargli che questo ingresso è terminato con il comando **RETURN**, il che gli restituirà la mano.
Se l'informazione inserita non è di natura adeguata (battete, per esempio, un testo su un INPUT X, che aspetta un numero), il messaggio **REDO FROM START** è visualizzato, e il calcolatore si rimette in attesa.
È possibile, come per il PRINT, visualizzare dei messaggi prima dell'ingresso della variabile, ed introdurre dei comandi di movimento del cursore.
Potremo, per esempio, scrivere:

INPUT "IL TUO NOME,PER FAVORE" ; A$

L'ISTRUZIONE GOTO, O ISTRUZIONE DI RINVIO INCONDIZIONATO

Una delle particolarità del linguaggio Basic, che lo rende facilmente accessibile, è che ***ogni linea di programma deve essere preceduta da un numero***: queste linee sono in seguito eseguite dal personal computer in sequenza, nel senso crescente dei numeri. Per diverse ragioni, potrebbe essere necessario, durante lo svolgimento di un programma, di non seguire questo ordine sequenziale molto rigido; utilizzeremo allora ***l'istruzione GOTO, che permette di dirottare il funzionamento del programma dal suo corso normale***, con la possibilità d'altronde di riprenderlo in seguito con un altro GOTO. Il rinvio deve, peraltro, imperativamente farsi su un numero di linea esistente nel programma (GOTO 20, per esempio).
Per dare un po' di elasticità a questa istruzione di rinvio, esiste un'istruzione molto potente, che permette di rinviare ad un qualunque numero di linea, secondo il valore di una variabile numerica: **è l'istruzione ON|____|GOTO|____|**; scriveremo, per esempio:

ON X GOTO 100, 200, 300 ed il rinvio sarà fatto alla linea 100 se X vale 1, alla linea 200 se X vale 2, e 300 se X vale 3.

L'ISTRUZIONE IF|___|THEN|___| , O ISTRUZIONE DI TEST

Possiamo utilizzarla in due maniere diverse:
— In rottura di sequenza condizionale, sotto la forma

IF |condizione = vera| THEN |numero di linea|

Quando dopo la parola **THEN** viene messa l'istruzione **GOTO**, non è necessario che la parola **THEN** venga espressa. Essa è invece necessaria con **GOSUB**.
È l'utilizzo più corrente di questo test, che permette di rinviare ad un'altra parte del programma, se per esempio X=0, o se A=B, ecc.; gli **operatori logici NOT**, **OR** o **AND** possono d'altronde essere utilizzati nella scrittura della condizione.
Potremo così utilizzare il testo sotto la forma:

IF |condizione| THEN |istruzione|

l'istruzione presentata al seguito di THEN può essere un'assegnazione A=12, un ingresso INPUT B$, o anche un altro test IF|___|THEN|___|.

Possiamo per esempio scrivere:

IF A>3 THEN IF B<7 THEN PRINT A+B

L'ISTRUZIONE FOR ...NEXT, O ISTRUZIONE DI CICLI

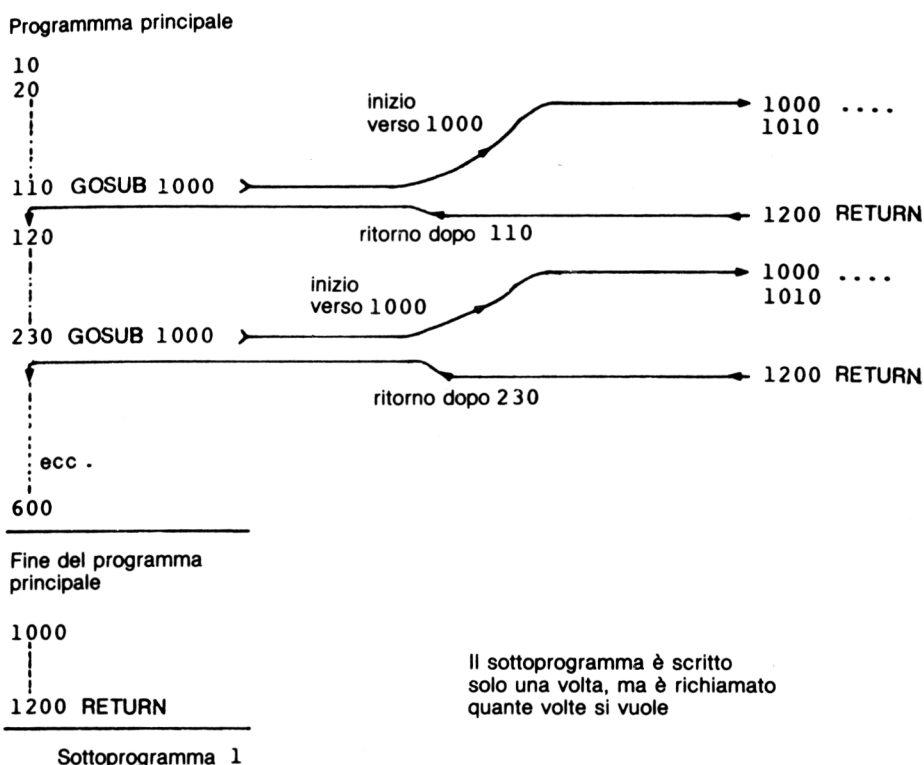
Siamo spesso condotti, in programmazione, a voler eseguire un numero intero di volte una certa serie d'istruzioni (per esempio, fare avanzare un punto mobile dalla casella 6 alla casella 20, ecc.). Piuttosto che utilizzare un test ed un rinvio, sarà meglio utilizzare un ***CICLO***, la variabile di ciclo prendendo i suoi valori negli interi o nei decimali, ed il ***passo*** di questa variabile (cioè la quantità di cui aumenta o diminuisce ad ogni passaggio) può anch'esso prendere i suoi valori negli interi o nei decimali.
Così se scriviamo:

```
FOR I =   } 0 TO 20 STEP 2
NEXT I    } istruzioni
```

le istruzioni contenute nel ciclo saranno eseguite per I=0, per I=2,... e questo fino a I=20, ossia 11 volte. La variabile I può essere usata per fare un calcolo nel ciclo, ma non dovrà esservi modificata. Possiamo, naturalmente, uscire dal ciclo prima di I=20, con un test portante una quantità qualunque.
Se ***il passo del ciclo (STEP)*** non è precisato, sarà automaticamente preso uguale a 1.

È possibile concatenare vari cicli di nome diverso, ed omettere l'indicazione della variabile nel NEXT.

L'ISTRUZIONE GOSUB, O L'ISTRUZIONE DI RICHIAMO DI SOTTOPROGRAMMA

Quando una stessa serie di istruzioni deve essere eseguita varie volte, e richiamata da diverse zone del programma, eviteremo di riscriverla varie volte. La scriveremo una volta per tutte, e la posizioneremo fuori del programma vero e proprio (dandole, per esempio, un numero di linea superiore al numero dell'ultima linea del programma). Indipendentemente dalle difficoltà che potremo incontrare al livello del nome delle variabili, basterà chiamarla con **l'istruzione GOSUB** **numero di linea**. Il programma farà allora una deviazione nella parte del programma richiamata, ma non potrà ritornare e proseguire il suo svolgimento normale se incontra nel programma richiamato un'istruzione **RETURN**, o istruzione di ritorno.
Ecco lo schema teorico di funzionamento:

Un sottoprogramma potrà a sua volta eventualmente richiamarne un altro, e così via.

Il numero misterioso

RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: gioco di deduzione per un giocatore.
Difficoltà del gioco: facile.

IL GIOCO: Si chiede al calcolatore di creare un numero di n cifre (con n compreso tra 1 e 7). Si dovrà quindi cercare di scoprirlo, col minor numero di tentativi possibile, ricevendo le sole informazioni TROPPO PICCOLO o TROPPO GRANDE.

IL PROGRAMMA: La linea 40 permette di creare il numero da indovinare.
Gli inputs sono fatti dalle linee 60 e 70.
La linea 80 controlla se il risultato è esatto.
La linea 90 segnala un tentativo troppo piccolo, e la linea 100 uno troppo grande.
Il contatore delle prove è alla linea 110.
Le linee da 200 a 240 si occupano della visualizzazione dei risultati, e della gestione della parte che segue.

PER GIOCARE: Si farà partire il programma con RUN.
Si otterrà un primo messaggio NUMERO DI CIFRE? al quale si risponde con un intero tra 1 e 7, seguito da RETURN.
Si avrà allora il messaggio PROVA N. 1, seguito dalla domanda IL TUO NUMERO. Si inserisce un numero e si schiaccia RETURN; si ottiene allora la risposta TROPPO PICCOLO o TROPPO GRANDE, oppure, se si è riusciti ad indovinare, il messaggio HAI TROVATO———IN———PROVA. Se il risultato non è esatto, si ritorna a PROVA N. 2, e IL TUO NUMERO.
Alla fine del gioco, appare la domanda UN'ALTRA PARTITA? (s/n); si risponde s per fare un'altra partita (seguito da RETURN).
Si può fermare il gioco in qualsiasi momento battendo RESTORE.

ESTENSIONI POSSIBILI: Potremmo immaginare di fare indovinare al giocatore non un numero ma qualcos'altro (per esempio una parola, utilizzando l'ordine alfabetico, o un'altra cosa ancora...).

```
00010 PRINT"Il NUMERO Misterioso 001@&"
00020 PRINT:INPUT "Numero CIFRE :";N
00030 IFN>7THENRUN
00040 X=INT(RND(1)*10 N+1)
00050 J=1
00060 INIT 7
00070 PRINT:PRINT"PROVA Num.";J
00080 INPUT "Il tuo NUMERO :";M
00090 IFM=XTHEN200
00100 IFM<XTHENPRINT"Troppo PICCOLO"
00110 IFM>XTHENPRINT"Troppo GRANDE"
00120 J=J+1
00130 PRINT:PRINT
00140 GOTO 70
00200 INIT 7
00210 PRINT"Hai TROVATO ";X;" in ";J;
00220 PRINT" PROVE !":PRINT
00230 INPUT "Un'altra PARTITA(s/n):";R$
00240 IFR$="s"THENPRINT:RUN
```

```
PROVA Num. 4
Il tuo NUMERO :? 95
Troppo PICCOLO


PROVA Num. 5
Il tuo NUMERO :? 99
Troppo PICCOLO


PROVA Num. 6
Il tuo NUMERO :? 100
Hai TROVATO  100 in  6 PROVE !

Un'altra PARTITA(s/n):? s
```

Il cubo

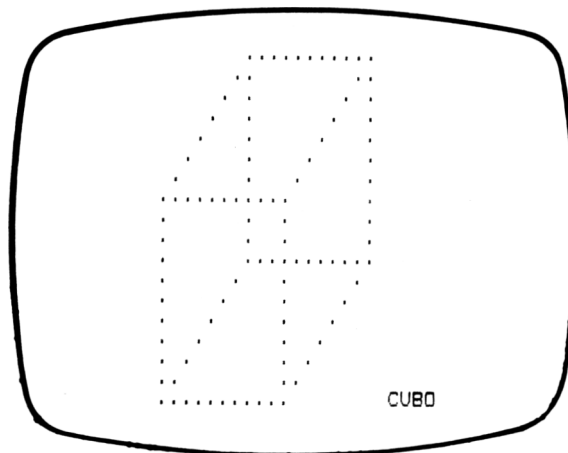
RIFERIMENTI:
Difficoltà di programmazione: nessuna.
Categoria del gioco: disegno.
Difficoltà del gioco: nessuna.

IL GIOCO:
Il programma proposto si contenta di realizzare con l'aiuto dell'istruzione PRINT una vista di un cubo in prospettiva cavaliera.
Potrebbe essere per esempio incorporato in un gioco basato sulle capacità di riconoscimento delle forme geometriche elementari.

IL PROGRAMMA:
Il disegno del cubo è realizzato dalla stampa di punti, utilizzando due cicli: il ciclo (30-120), e il ciclo (130-180).
Il nome dell'oggetto raffigurato è scritto alla linea 190 (qui: CUBO).

PER GIOCARE:
Basta far partire il programma con RUN, e guardare.

ESTENSIONI POSSIBILI:
L'alta risoluzione permetterà di avere un tracciato più preciso.
Potremmo immaginare un gioco utilizzante dei parallelepipedi di differenti misure, con visualizzazione per esempio dei tre lati, dovendo indovinare il volume, ecc.

```
00010 REM CUBO
00020 INIT 7 :PAGE
00030 FOR I=1 TO 10
00040 CURSORX 7+I:CURSORY 0:PRINT"."
00050 CURSORX I:CURSORY 7:PRINT"."
00060 CURSORX 7+I:CURSORY 10:PRINT"."
00070 CURSORX I:CURSORY 17:PRINT"."
00080 CURSORX 1:CURSORY 6+I:PRINT"."
00090 CURSORX 11:CURSORY 6+I:PRINT"."
00100 CURSORX 8:CURSORY I:PRINT"."
00110 CURSORX 18:CURSORY I:PRINT"."
00120 NEXT I
00130 FOR I=1 TO 8
00140 CURSORX I:CURSORY 8-I:PRINT"."
00150 CURSORX 10+I:CURSORY 8-I:PRINT"."
00160 CURSORX I:CURSORY 18-I:PRINT"."
00170 CURSORX 10+I:CURSORY 18-I:PRINT".
00180 NEXT I
00190 CURSORX 20:CURSORY 17:PRINT"CUBO"
```

Caldo, freddo

RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Uno dei più vecchi giochi del mondo, sempre in voga, ma questa volta è il calcolatore che vi fa da compagno: infatti nasconderà un oggetto in una certa posizione di una griglia di cui avrete voi stessi scelto le dimensioni. Per trovarlo, non vi darà che qualche indicazione, sotto la forma BRUCI, o FA FREDDO, ecc. Dovete riuscire a trovare l'oggetto nascosto, col minor numero di tentativi possibile (il vostro punteggio sarà visualizzato alla fine della partita).

IL PROGRAMMA: Le linee 50 e 60 definiscono la posizione dell'oggetto.
La linea 110 calcola la distanza, in funzione delle due coordinate che avete inserito alle linee 80-90.
La linea 120 controlla se avete vinto.
Le linee 150-160 normalizzano la distanza (che deve avere un senso, quali che siano le dimensioni della griglia iniziale).
La linea 140 effettua la partenza, verso uno dei sottoprogrammi di visualizzazione.

PER GIOCARE: Si fa partire il programma con RUN, poi si inseriscono le dimensioni della griglia (quadrata) e si batte RETURN.
Già dalla visualizzazione seguente (Dammi X) si inserisce la coordinata supposta di X, seguita da RETURN. Si ottiene in seguito il messaggio (Dammi Y), e si inserisce Y, seguito da RETURN.
Si ottiene in risposta un testo, che varia in funzione della distanza dall'oggetto cercato (si va da COMPLETAMENTE GHIACCIATO a BOLLENTE). Quando lo si è trovato, si ottiene la visualizzazione del numero di tentativi effettuati.
Per uscire dal programma prima della fine della partita, basta premere RESTORE.

ESTENSIONI POSSIBILI: Se si dispone del colore, sarà interessante trovare un modo più visivo di simbolizzare la distanza, per esempio delle mattonelle varianti dal rosso al blu, nelle posizioni tentate.
Si potrebbe anche lavorare in uno spazio a tre dimensioni, nascondere vari oggetti, ecc.

```
00010 PRINT"CALDO, FREDDO 003@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INPUT "Dimensione GRIGLIA :";D
00050 X=INT(RND(1)*D+1)
00060 Y=INT(RND(1)*D+1)
00070 INIT 7
00080 INPUT "Dammi X :";A
00090 INPUT "Dammi y :";B
00100 S=S+1
00110 T=ABS(X-A)+ABS(Y-B)
00120 IFT=0THEN700
00130 PRINT"Sei ";
00150 N=INT(SQR(D)/T)+1
00160 IFN>5THENN=5
00170 IFT>SQR(D)THEN190
00180 ONNGOTO 200,300,400,500,600
00190 PRINT"Completamente GHIACCIATO"
00195 GOTO 80
00200 PRINT"Appena TIEPIDO"
00210 GOTO 80
00300 PRINT"TIEPIDO"
00310 GOTO 80
00400 PRINT"CALDO"
00410 GOTO 80
00500 PRINT"Molto CALDO"
00510 GOTO 80
00600 PRINT"BOLLENTE"
00610 GOTO 80
00700 INIT 7
00710 PRINT"Hai TROVATO in ";S;"COLPI"
```

```
Sei Completamente GHIACCIATO
Dammi X :? 22
Dammi y :? 2
Sei Completamente GHIACCIATO
Dammi X :? 2
Dammi y :? 2
Sei Completamente GHIACCIATO
Dammi X :? 2
Dammi y :? 3
Sei Completamente GHIACCIATO
Dammi X :? 2
Dammi y :? 4
Sei TIEPIDO
Dammi X :? 3
Dammi y :? 4
Sei CALDO
Dammi X :? 4
Dammi y :? 4
Hai TROVATO in  12COLPI
```

I fiammiferi

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: contro il Videopac.
Difficoltà del gioco: media.

IL GIOCO:

Il giocatore dovrà all'inizio della partita scegliere un certo numero di fiammiferi, così come il numero massimo di fiammiferi da prendere ad ogni mano.
Decide quindi se cominciare, o se lasciare iniziare il Videopac. Ognuno prende a turno tanti fiammiferi quanti ne desidera, nel limite del massimo consentito. Quello che prende l'ultimo fiammifero è considerato come perdente.

IL PROGRAMMA:

La mano del giocatore comincia alla linea 70. Quella del Videopac alla linea 110. La linea 80 controlla la validità del tiro del giocatore.
La linea 90 gestisce il cambiamento del giocatore.
Il tiro del Videopac è calcolato in un ciclo non formale, dalla linea 120 alla linea 150. Il risultato è visualizzato dalla linea 1020.
Se la partita è terminata, la linea 1030 lo percepisce e rinvia alla linea 2000 per il messaggio finale.

PER GIOCARE:

Si parte con RUN, e, all'apparizione del messaggio Numero TOTALE, si inserisce il numero di fiammiferi, seguito da RETURN. Poi si inserisce il numero massimo autorizzato, seguito da RETURN. Se il giocatore vuole iniziare, batte in seguito 0; altrimenti batte 1, poi RETURN. Dopo ogni mano, il numero di fiammiferi rimanenti è visualizzato, e il Videopac visualizza TU TIRI quando tocca al giocatore prendere. Chi prende l'ultimo fiammifero ha perso.

ESTENSIONI POSSIBILI:

Potremmo, naturalmente, immaginare una visualizzazione grafica a colori più spettacolare; potremmo anche lasciare al caso la scelta del numero massimo, ecc.

```
00010 PRINT"I fiammiferi 0040&"
00020 INIT 7:CLEAR
00030 INPUT "Numero TOTALE :";N
00040 INPUT "PRESA Massima :";I
00050 INPUT "Chi inizia (Tu=0/io=1):";D
00060 IFD=1GOTO 110
00070 INPUT "Tu tiri :";P
00080 IFP<>INT(P) ORP<1 ORP>NGOTO 70
00090 D=1
00100 GOTO 1000
00110 S=I+D
00120 T=(N-S)/(I+D)
00130 IFT=INT(T)GOTO 200
00140 S=S-D
00150 GOTO 120
00200 P=ABS(S-D)
00210 IFP=0THENP=D
00220 D=0
01000 PRINTP
01010 N=N-P
01020 PRINT"Rimane ";N
01030 IFN=0GOTO 2000
01040 GOTO 60
02000 INIT 7
02010 IFD=1THENPRINT"Ho vinto":SOUND 0
02020 IFD<>1THENPRINT"BRAVO":SOUND 4
```

```
run30
Numero TOTALE :? 23
PRESA Massima :? 7
Chi inizia (Tu=0/io=1):? 0
Tu tiri :? 6
 6
Rimane  17
 1
Rimane  16
Tu tiri :? 7
 7
Rimane  9
 1
Rimane  8
Tu tiri :? 7
 7
Rimane  1
 1
Rimane  0
```

I fiammiferi in molti

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: logica (vari giocatori).
Difficoltà del gioco: abbastanza difficile.

IL GIOCO:

In questo gioco per vari giocatori, il direttore del gioco dovrà, all'inizio della partita, scegliere un numero di fiammiferi, ed indicare al Videopac il numero di giocatori.
Ad ogni mano, il giocatore che ha il tiro è autorizzato a riprendere dal mucchio di fiammiferi rimanenti, fino al doppio di quanto ha preso il giocatore precedente; colui che prende l'ultimo fiammifero ha vinto (o perso se si modificano leggermente le regole).

IL PROGRAMMA:

Le linee da 10 a 40 permettono i differenti ingressi.
Le linee da 70 a 100 realizzano le stampe.
La linea 120 controlla il tiro di ogni giocatore (in rapporto al tiro precedente).
La linea 180 gestisce le mani (rimessa a 1 del numero del giocatore).
La linea 160 fa il test di fine, e le linee 210-220 visualizzano il numero del giocatore vincente (con la regola prevista inizialmente).

PER GIOCARE:

Dopo aver lanciato il programma con RUN, si inserisce il numero di fiammiferi iniziale, seguito da RETURN, poi il numero di giocatori (e RETURN).
Il primo giocatore potrà prendere solo un fiammifero, il secondo ne potrà prendere uno o due; il seguente 1, 2, o 1, 2, 3 o 4 in funzione della mano precedente, e così via; il numero massimo è ogni volta visualizzato dal Videopac.
Quando l'ultimo fiammifero è stato preso, si ottiene il messaggio IL GIOCATORE. . .ha VINTO, e la partita è terminata.

ESTENSIONI POSSIBILI:

Si può decidere che quello che prende l'ultimo fiammifero è eliminato dal gioco, e ricominciare la partita con i giocatori rimanenti, che saranno anch'essi eliminati uno dopo l'altro.
Possiamo anche immaginare una rappresentazione grafica del mucchio di fiammiferi, fissare a caso il numero di fiammiferi da prendere all'inizio della partita, ecc.

```
00010 PRINT"FIAMMIFERI in Molti 005@&"
00020 FOR I=1 TO 500:INIT 7
00030 INPUT "Num.TOTALE FIAMMMIFERI:";N
00040 INPUT "Num. GIOCATORI :";J
00050 F=1
00060 K=1
00070 INIT 7
00080 PRINT"Turno del GIOCATORE :";K
00090 PRINT"TOTALE che rimane :";N
00100 PRINT"MASSIMO da Togliere :";F
00110 INPUT "Tuo TIRO :";C
00120 IFC=0ORC>FTHEN110
00130 N=N-C
00140 IFN<0THENN=N+C:GOTO 110
00150 F=C+C
00160 IFN=0THEN200
00170 K=K+1
00180 IFK>JTHENK=1
00190 GOTO 70
00200 INIT 7
00210 PRINT"Il GIOCATORE Num.";K;
00220 PRINT" ha VINTO !!!"
```

```
0121345678 9ABCDEFGHIJKLMNOPQRSTUVWXYZ

1

Giocatore  1 - tuo COLPO : 1

Premere -RET- ?
```

Calendario perpetuo

RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: gadget pubblicitario.
Difficoltà del gioco: nessuna.

IL GIOCO: Questo programma vi permetterà di creare tutti gli anni il vostro calendario; potrete così servirvene per mandare ai vostri amici un calendario del mese della loro nascita, o per augurar loro buone feste, ecc.

IL PROGRAMMA: Le linee 30 e 40 vi permettono di inserire il mese e l'anno.
Le linee 50, 60, 70 e 80 calcolano il numero del primo giorno del mese (non si è tenuto conto qui del 29 febbraio, per semplificare).
Il ciclo A (130-160) realizza la stampa del calendario.

PER GIOCARE: Dopo aver lanciato il programma con l'istruzione RUN, si dovrà introdurre l'anno, sotto forma, per esempio, 1983, seguito da RETURN; si ottiene allora la domanda MESE, alla quale conviene rispondere con il numero del mese, secondo l'ordine usuale (1 per gennaio, 2 per febbraio, ecc.).
Si batte in seguito RETURN, e il calendario è allora visualizzato sullo schermo.

ESTENSIONI POSSIBILI: Potremo senza difficoltà prevedere la stampa dei 29 febbraio, prevedere un'opzione che permetta la stampa per un anno completo, prevedere l'introduzione del nome dei mesi in chiaro, ecc.

```
00010 PRINT"CALENDARIO PERPETUO"
00020 FOR I=1 TO 500:INIT 7
00030 INPUT "Per che ANNO :";A
00040 INPUT "Per che MESE :";M
00050 D=INT(365.25*(A+(M<3)))+1
00060 T=INT((INT((A+(M<3))/100)-7)*.75)
00070 D=D+INT(30.6*(M+1-(M<3)*12))-T
00080 D=D-7*INT(D/7)
00090 N=(M=4ORM=6ORM=9ORM=11)+(M=2)*3
00100 M=31+N:PRINT
00110 PRINT"DOM LUN MAR MER GIO VEN";
00120 PRINT" SAB":PRINT
00130 FOR A=1 TO M
00140 PRINTTAB(4*(D+A-1));A:
00150 IFD+A-1=6THENPRINT:D=D-7
00160 NEXT A
```

```
Per che ANNO :? 1984
Per che MESE :? 12

DOM LUN MAR MER GIO VEN SAB

                         1
 2   3   4   5   6   7   8
 9   10  11  12  13  14  15
 16  17  18  19  20  21  22
 23  24  25  26  27  28  29
 30  31
```

Battaglia navale

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: riflessione.
Difficoltà del gioco: facile.

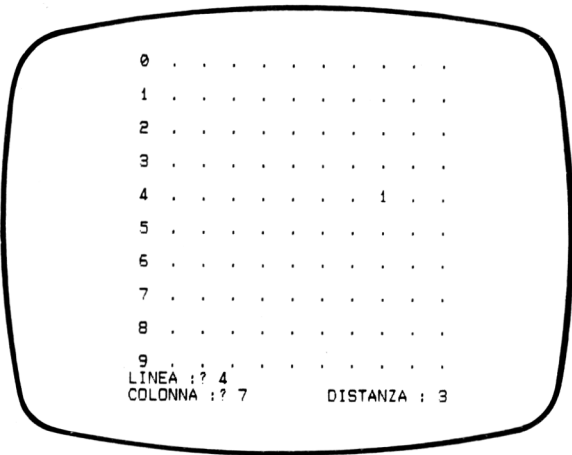
IL GIOCO:

Una battaglia navale a priori abbastanza semplice, con una sola nave nascosta a caso in una griglia di 10 * 10 caselle.
Ad ogni colpo, bisognerà tirare dando le coordinate H (orizzontale) e V (verticale) della casella mirata, e il Videopac risponde visualizzando la distanza che separa la nave invisibile dalla casella mirata. Ma, se si tira troppo vicino, la nave potrebbe rinascondersi a caso.

IL PROGRAMMA:

Il ciclo I (40-70) disegna il campo di gioco.
Le linee 90-100 definiscono la posizione della nave.
Le coordinate di tiro sono inserite alle linee 110 e 120, e la linea 130 visualizza il colpo sullo schermo. La distanza è calcolata alla linea 140, e la linea 150 guarda se si è fatto o no cilecca.
La linea 170 interviene per spostare la nave se i colpi si avvicinano troppo.
La linea 180 permette la visualizzazione della distanza.

PER GIOCARE:

Il campo di gioco è visualizzato dopo aver battuto RUN. Al messaggio LINEA, si risponde dando il numero della linea orizzontale sulla quale si desidera tirare. Si batte quindi RETURN, e, alla domanda COLONNA, si risponde dando il numero della colonna verticale seguito da RETURN.
Se si tira troppo vicino, si ottiene il messaggio PFFFF... che significa che la nave nemica ha cambiato posizione.
Se si tira sulla casella giusta, si ottiene la visualizzazione del numero totale dei tiri effettuati per vincere.

ESTENSIONI POSSIBILI:

Sono innumerevoli, anche perché la battaglia navale è uno dei giochi più famosi che esistano. Possiamo introdurre due giocatori, giocare contro il Videopac, mettere varie navi, dei sottomarini, ecc.

```
00010 PRINT"BATTAGLIA NAVALE 007@&"
00015 INPUT "Dammi un INTERO :":F
00020 FOR I=1 TO F=Y=RND(1):NEXT I
00025 INIT 7:PAGE:STORE:CURSORY 0
00030 PRINT"    0  1  2  3  4  5  6  ";
00035 PRINT"7  8  9"
00040 FOR I=0 TO 9
00050 PRINTI;"  .  .  .  .  .  .  .  ";
00060 PRINT".  .  .":PRINT
00070 NEXT I:SCREEN:DISPLAY
00080 S=1
00090 X=INT(RND(1)*10)
00100 Y=INT(RND(1)*10)
00110 CURSORY 20:INPUT "LINEA :";H
00120 INPUT"COLONNA :";V
00130 CURSORX3*V+4:CURSORY2*H+1:PRINTS
00140 D=INT(SQR((X-H) 2+(Y-V) 2))
00150 IFD=0THEN200
00160 S=S+1
00165 CURSORX 21:CURSORY 21
00170 IFD=1THEN:PRINT"PFFFF...":GOTO220
00180 PRINT"DISTANZA :";D;"   ":GOTO 90
00190 GOTO 90
00200 INIT 7
00210 PRINT"TROVATO in ";S;" COLPI":END
00220 PRINT"                   ":GOTO90
```

```
0  .  .  .  .  .  .  .  .  .  .

1  .  .  .  .  .  .  .  .  .  .

2  .  .  .  .  .  .  .  .  .  .

3  .  .  .  .  .  .  .  .  .  .

4  .  .  .  .  .  .  .  1  .  .

5  .  .  .  .  .  .  .  .  .  .

6  .  .  .  .  .  .  .  .  .  .

7  .  .  .  .  .  .  .  .  .  .

8  .  .  .  .  .  .  .  .  .  .

9  .  .  .  .  .  .  .  .  .  .
LINEA :? 4
COLONNA :? 7        DISTANZA : 3
```

Golf

RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: solitario.
Difficoltà del gioco: abbastanza facile.

IL GIOCO: Si tratta di riuscire a fare, col minor numero di colpi possibile, un percorso di golf con 18 buche. Per fare ciò, occorre colpire la palla con una forza più o meno grande, fino ad ottenere la sua caduta nella buca.
Ma attenzione, il caso potrebbe intervenire su ogni colpo; così spesso dovrete tornare indietro (bisognerà allora far precedere l'indicazione della forza con un segno —).

IL PROGRAMMA: La linea 70 permette di definire la posizione della buca, e le linee (75÷85) disegnano il campo di gioco.
Lo spostamento da realizzare è calcolato alle linee 120 e 125.
Il ciclo I (140÷190) realizza lo spostamento della palla.
La linea 200 guarda se il colpo è, o no, vincente; se lo è, si ricomincia (linea 250) fino a che si realizzino 18 buche (ciclo S (40÷250)); fine del ciclo in 270, con visualizzazione della prestazione.

PER GIOCARE: Si ottiene la visualizzazione del campo con RUN, così come il disegno della palla nella sua posizione iniziale, ed il testo: BUCA 1. FORZA; si deve allora rispondere dando la forza desiderata (un numero tra 1 e 300...).
Subito dopo aver battuto RETURN, la palla ricomincia a spostarsi; può a questo punto sia fermarsi davanti alla buca (nel qual caso si ricomincia), sia finirci dentro (si passa allora alla buca seguente), sia sorpassarla: si dovrà allora batterla con una forza negativa. Il gioco finisce con la scritta 18 BUCHE in ... tiri.

ESTENSIONI POSSIBILI: Potremmo espandere il gioco per fare partecipare più giocatori, migliorare la qualità grafica, utilizzare una rappresentazione spaziale del campo di gioco, avere un terreno più accidentato, degli ostacoli, cambiare il club...

```
00010 PRINT" *  G O L F  *  008@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 FOR S=1 TO 18
00050 INIT 7
00060 X=1
00070 T=INT(29*RND(1)+9)
00075 CURSORY 5
00080 FORJ=1TO38:PRINTCHR$(126);:NEXTJ
00085 CURSORX T:CURSORY 5:PRINT" "
00090 CURSORX X:CURSORY 4:PRINT"O"
00100 CURSORX 15:CURSORY 8
00105 PRINT"                    "
00110 CURSORY8:PRINT"BUCA :";S;
00115 INPUT "          FORZA :";R
00120 F=ABS(R)*(SQR(RND(1))+.1) 2
00125 F=INT(R*.16-F/1000)
00130 IF F=0 THEN 100
00140 FOR I=SGN(F) TO F STEP SGN(F)
00150 X=X+SGN(F)
00160 IFX<39ANDX>1THEN280
00170 IFX>38THENX=38
00180 IFX<=1THENX=1
00190 NEXT I
00200 IFX=TTHEN230
00210 E=E+1
00220 GOTO 75
00230 CURSORX X:CURSORY 4:PRINT" "
00235 CURSORX X:CURSORY 4:PRINT"O"
00240 FOR P=1 TO 500:NEXT P
00250 NEXT S
00260 INIT 7
00270 PRINT"18 BUCHE in ";E;" Tiri":END
00280 CURSORXX-SGN(F):CURSORY4:PRINT" "
00290 CURSORX X:CURSORY 4:PRINT"O"
00300 GOTO 170
```

```
                O
~~~~~~~~~~~~~~~~ ~~~~~~~~~~~~~~~~~~~~~

BUCA : 1        FORZA :? 7
```

Solfeggio

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessione.
Difficoltà del gioco: facile.

IL GIOCO:

Questo programma vi propone un'iniziazione elementare al solfeggio, permettendovi di imparare a riconoscere facilmente le differenti note, che appaiono le une dopo le altre su uno spartito; dopo la visualizzazione della nota, dovete inserire il suo nome in chiaro ed il vostro punteggio sarà visualizzato in permanenza. Il programma è inizialmente previsto per la chiave di sol, ma sarà facilmente adattabile ad un'altra chiave.

IL PROGRAMMA:

Il ciclo I (60÷80) disegna uno spartito.
La nota è scelta alla linea 80 e visualizzata alle linee 90, 100 e 110.
La linea 125 visualizza il punteggio.
La linea 130 visualizza il nome della chiave (qui chiave di sol) e vi interroga.
La linea 140 si occupa della fine della partita.
Le linee (150÷170) effettuano la decodificazione della nota inserita.
La linea 200 controlla il risultato.

PER GIOCARE:

Dopo aver lanciato il programma con RUN, lo spartito apparirà, così come la prima nota. Risponderete alla domanda NOTA? dandone il nome in chiaro (do, re, mi, fa, sol, la, si), seguito da RETURN. Il punteggio vi indicherà se è esatto o no, ed una nuova nota sarà visualizzata. Per fermare, basterà rispondere 0 seguito da RETURN, alla domanda NOTA?

ESTENSIONI POSSIBILI:

Potremmo innanzitutto facilmente modificare il programma per permettervi di lavorare con le diverse chiavi. Potremmo anche migliorare la grafica, introdurre il colore, il suono (quando si risponde giusto si sentirà la nota, per esempio). Potremmo anche realizzare, seguendo lo stesso principio, un gioco che vi permetta di riconoscere degli accordi, ecc.

```
00010 PRINT"SOLFEGGIO 009@&"
00020 INPUT "Dammi un Num.INTERO :":F
00030 FOR I=1 TO F=Y=RND(1):NEXT I
00040 S=0:P=0:Z$=CHR$(125)
00050 INIT 7:PAGE:STORE
00060 FORI=1TO5:CURSORX 1:CURSORY 2*I+1
00070 FORJ=1TO38:PRINTCHR$(96);:NEXT J
00080 NEXT I:X=INT(RND(1)*12)
00090 CURSORX 18:CURSORY X:PRINTZ$
00100 CURSORX 18:CURSORY X+1:PRINTZ$
00110 CURSORX 18:CURSORY X+2:PRINT"O"
00120 CURSORX1:CURSORY18:SCREEN:DISPLAY
00125 PRINT"PUNTI ";S;" su";P;" PROVE"
00130 PRINT "Chiave di SOL - Che NOTA";
00135 INPUT " e' :";R$
00140 IFR$="0"THENSOUND 4:SOUND 2:END
00150 Y=(1ANDR$="fa")+(2ANDR$="mi")
00160 Y=(3ANDR$="re")+(4ANDR$="do")
00170 Y=(5ANDR$="si")+(6ANDR$="la")
00180 P=P+1
00190 IFR$="sol"THENR$="so"
00200 IFY=XORY=X-7THENS=S+1:SOUND 4
00210 GOTO 50
```

Blackjack

(Gioco del 21)

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: gioco contro il Videopac.
Difficoltà del gioco: facile.

IL GIOCO:

Bisogna inserire una puntata all'inizio del gioco, e potremo lanciare i dadi battendo 1; ad ogni lancio si otterrà un nuovo totale, che aumenta poco a poco, e si avvicinerà a 21. Si deve tentare di avvicinarsi il più possibile a 21, senza superare questo numero. Quando decideremo di fermarci, basterà battere 0 per lasciar giocare il Videopac che cercherà allora di ottenere un miglior punteggio. Alla fine di ogni partita, il guadagno sarà visualizzato così come il totale delle vincite (o delle perdite).

IL PROGRAMMA:

Le linee 50, 60 e 70 realizzano il controllo di quello che deve giocare (giocatore o BANCO).
Il sottoprogramma (400-410) lancia i dadi.
Le linee 110 e 120 calcolano il totale fatto dal giocatore e lo visualizzano.
Il totale del BANCO è calcolato alla linea 200.
La linea 220 controlla la fine del gioco per il BANCO.
Le linee 260 e 280 visualizzano i risultati, e la linea 300 cerca qual è il vincitore.
La vincita (o la perdita) sarà visualizzata dalla linea 320, ed il totale dalla linea 350.
La linea 380 controlla la fine della partita.

PER GIOCARE:

Battere RUN per lanciare il gioco.
Alla domanda PUNTATA, inserire la scommessa iniziale, seguita da RETURN, e battere 1 per lanciare il dado. Il primo risultato è visualizzato nel seguente modo Hai Totalizzato... e si dovrà ribattere 1 seguito da RETURN, finché si desidera ricominciare a tirare.
Appena finito, si battono 0 e RETURN per lasciar giocare il Videopac (che ha qui il ruolo di banca).
La vincita (o la perdita, se c'è il segno —) apparirà alla fine del gioco, così come il totale delle vincite (o delle perdite) realizzate dall'inizio.
Se si desidera interrompere il gioco, si dovrà battere n in risposta alla domanda SI CONTINUA (s/n).

ESTENSIONI POSSIBILI: Questo gioco sarà evidentemente più interessante se è fatto in tanti, e sarà facile modificare il programma di conseguenza; potremmo anche disegnare il dado o, meglio, utilizzare delle carte, il che permetterà allora di fare il vero BLACKJACK.

```
00010 PRINT"GIOCO del 21"
00020 G=0
00030 INPUT "Dammi un Num.INTERO :";F
00040 FOR I=1 TO F:Y=RND(1):NEXT I
00050 INIT 7
00060 TT=0
00070 INPUT "La tua Puntata :";M
00080 INPUT"con 1 tiri, con 0 esci :";R
00090 IFR=0GOTO 150
00100 GOSUB 400
00110 T=T+X
00120 PRINT"Il tuo Totale e' :";T
00130 GOTO 80
00150 INIT 7
00160 PRINT"*Il BANCO gioca *"
00170 S=0
00180 GOSUB 400
00190 S=S+X
00200 PRINT"Totale del BANCO :";S
00210 FOR I=1 TO 500:NEXT I
00220 IF S>T OR S>17 THEN GOTO 250
00230 GOTO 180
00250 INIT 7
00260 PRINT"Hai totalizzato :";T
00270 PRINT
00280 PRINT"Il BANCO ha fatt0 :";S
00290 PRINT
00300 IFS>21 OR (S<T AND T<=21)GOTO 320
00310 M=-M
00320 PRINT"Vincita :";M
00330 PRINT
00340 G=G+M
00350 PRINT"Il tuo TOTALE Attuale :";G
00360 PRINT
00370 INPUT "Si continua (s/n):";R$
00380 IFR$="n"THENEND
00390 GOTO 50
00400 X=INT(RND(1)*6+1)
00410 RETURN
```

```
La tua Puntata :? 50
con 1 tiri, con 0 esci :? 1
Il tuo Totale e' : 6
con 1 tiri, con 0 esci :? 1
Il tuo Totale e' : 10
con 1 tiri, con 0 esci :? 1
Il tuo Totale e' : 14
con 1 tiri, con 0 esci :? 1
Il tuo Totale e' : 17
con 1 tiri, con 0 esci :? 1
Il tuo Totale e' : 23
con 1 tiri, con 0 esci :? 0
```

Pari-dispari

RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: facile.

IL GIOCO: È un gioco di fortuna, per uno o più giocatori. Il Videopac lancia simultaneamente due dadi, non visibili, e fa tre domande: qual è la parità del totale, qual è il totale, ed infine dare il valore di uno dei dadi; si può in seguito proseguire il gioco o interromperlo in qualsiasi istante.

IL PROGRAMMA: I dadi sono tirati da un sottoprogramma posizionato alla linea 500.
La linea 80 tiene il conto del numero di partite, e le linee 180, 260, 290 contabilizzano i punti segnati.
La parità del totale è determinata dalle linee 130 e 140 (la linea 140 si paragona con la risposta del giocatore).
La linea 400 e seguenti realizzano il messaggio di fine partita: totale, numero di punti, numero di partite giocate. La linea 430 provoca l'arresto se si risponde n alla domanda "SI CONTINUA".

PER GIOCARE: Si lancia il programma con RUN. Basta in seguito rispondere alle diverse domande poste, seguendo scrupolosamente le indicazioni date e battendo RETURN dopo ogni risposta.
Quando si finisce una partita, si ottiene il totale attuale ed il numero di partite nella forma di, per esempio, 6 SU 10. Si ottiene allora la domanda "SI CONTINUA" (s/n) e si risponderà n seguito da RETURN per fermarsi.

ESTENSIONI POSSIBILI: Se si desidera giocare in tanti, sarebbe interessante predisporre la possibilità di scommettere e di aggiungere una gestione del punteggio per ciascuno dei giocatori. Potremmo anche prevedere una miglioria nella presentazione (utilizzo del colore, ecc.)

```
00010 PRINT"PARI - DISPARI 0110&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 T=0:R=0
00050 GOSUB 500
00060 A=X
00070 GOSUB 500
00080 R=R+1
00090 S=A+X
00100 INIT 7
00110 INPUT "Tot.PARI/DISPARI(0/1):";N
00120 PRINT
00130 P=S/2-INT(S/2)
00140 IFN=PORN-.5=PGOTO 170
00150 PRINT"FALSO"
00160 GOTO 190
00170 PRINT"ESATTO"
00180 T=T+1
00190 PRINT
00200 INPUT "TOTALE Presunto :";N
00210 IFN=SGOTO 250
00220 PRINT"PERSO"
00230 FOR I=1 TO 900:NEXT I
00240 GOTO 400
00250 PRINT
00260 T=T+2
00270 INPUT "BRAVO! Dammi un dado :";N
00280 IFN<>AANDN<>S-AGOTO 220
00290 T=T+3
00300 PRINT
00310 PRINT"BRAVO !!":GOTO 230
00400 INIT 7
00410 PRINT"Hai un TOTALE di :";T;
00420 PRINT" su ";R
00430 PRINT
00440 INPUT "Si continua (s/n) :";R$
00450 IFR$="n"THENEND
00460 GOTO 50
00500 X=INT(RND(1)*6+1):RETURN
```

```
Tot.PARI/DISPARI(0/1):? 0

FALSO

TOTALE Presunto :? 6
PERSO
```

Indovina un numero

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: facile.

IL GIOCO:

È un gioco di fortuna, di concetto molto semplice, ciò nonostante molto interessante: il giocatore introduce la sua puntata, e scommette su un numero, da 1 a 6; tre dadi sono allora lanciati; se il numero puntato appare una volta, il giocatore vince la sua scommessa. Se appare due volte raddoppia la sua puntata; se appare tre volte, la triplica.

IL PROGRAMMA:

Il lancio dei dadi è realizzato da un sottoprogramma posizionato alle linee 600 e 610.
La visualizzazione è realizzata dalle linee 150 e 165.
Le linee da 180 a 210 calcolano il guadagno nelle diverse configurazioni possibili.
La totalizzazione è effettuata dalla linea 230 per i guadagni, e dalla linea 510 per le perdite.
La variabile R$, testata alla linea 550, determina il proseguimento eventuale del gioco.

PER GIOCARE:

Il lancio del programma con RUN fa apparire un primo messaggio LA TUA SCOMMESSA?; si risponde con un numero, seguito da RETURN. Si ottiene allora un secondo messaggio DATE UN NUMERO DA 1 A 6; si inserisce un numero seguito da RETURN. Si ottiene allora un secondo messaggio DAMMI UNA CIFRA DA 1 A 6; si inserisce un numero seguito da RETURN; i tre dadi, raffigurati dalle lettere A, B, C appaiono allora, così come un messaggio indicante il vostro guadagno o la vostra perdita. Poi un'indicazione sul vostro totale attuale appare, così come la domanda "Si continua" (s/n)?; battere s per continuare, n per fermarsi (seguito da RETURN).

ESTENSIONI POSSIBILI:

Potremmo facilmente introdurre dei controlli (puntata minima, colpo proposto da 1 a 6, ecc.). Potremmo anche disegnare dei dadi con una grafica più elaborata, introdurre la possibilità di giocare in molti, o anche fare partecipare il Videopac.

```
00010 PRINT"Indovina una CIFRA 012@&"
00015 PRINT
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 G=0
00050 INIT 7
00060 INPUT "La tua POSTA :";N
00070 PRINT
00080 INPUT"DI' una CIFRA da 1 a 6 :";D
00090 GOSUB 600
00100 A=X
00110 GOSUB 600
00120 B=X
00130 GOSUB 600
00140 C=X
00150 PRINT:PRINT:PRINT"   a          b";
00155 PRINT"          c"
00160 PRINT:PRINT"   ";A;"          ";B;
00165 PRINT"          ";C
00170 T=0
00180 IFD=ATHENT=T+N
00190 IFD=BTHENT=T+N
00200 IFD=CTHENT=T+N
00210 IFT=0GOTO 500
00220 PRINT:PRINT:PRINT"Vinci !;t
00230 G=G+T
00240 GOTO 520
00500 PRINT:PRINT:PRINT"PERSO    "
00510 G=G-N
00520 PRINT:PRINT:PRINT"TOTALE ";
00525 PRINT" Attuale :";G
00530 PRINT
00540 INPUT "Si continua (s/n) :";R$
00550 IFR$="s"GOTO 50
00560 PRINT"GRAZIE e Arrivederci":END
00600 X=INT(RND(1)*6+1)
00610 RETURN
```

```
La tua POSTA :? 500

DI' una CIFRA da 1 a 6 :? 5


   a          b          c

   5          6          2


Vinci !;t


TOTALE  Attuale : 500

Si continua (s/n) :? s
```

Il gioco di Wytthof

RIFERIMENTI:
Difficoltà di programmazione: media.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: media.

IL GIOCO:
Disponiamo di una griglia quadrata di 10*10 caselle. Un punto appare al di fuori della diagonale e i giocatori lo dovranno spostare ad ogni turno, seguendo solamente tre direzioni: destra, basso e diagonale in basso a destra. Il primo giocatore che arriva alla casella (9,9) sarà considerato come vincente, o perdente, secondo la convenzione presa all'inizio.

IL PROGRAMMA:
Le linee 40, 50, 60 permettono di definire una posizione di partenza, al di fuori della diagonale.
La linea 80 e il ciclo I disegnano la griglia.
La linea 100 stampa la posizione del punto mobile.
Il test di fine è effettuato alla linea 140.
Il cambiamento del giocatore è assicurato dalla linea 190.
Le direzioni di spostamento disponibili, 1, 2 e 3, sono decodificate da tre sottoprogrammi che cominciano alle linee 300, 400, 500.

PER GIOCARE:
Battere RUN per lanciare l'esecuzione del programma; il campo di gioco appare, così come la posizione di partenza.
Poi si ottiene il messaggio GIOCATORE 1 e sotto la domanda DIREZIONE (1, 2, 3), alla quale si risponde battendo uno di questi tre numeri, seguito da RETURN.
Sarà in seguito il turno del giocatore 2. Il gioco termina quando uno dei giocatori arriva alla casella (9,9).

ESTENSIONI POSSIBILI:
Possiamo innanzitutto migliorare i controlli (l'uscita dal terreno non è impedita, per esempio), poi la visualizzazione (grafica, colori), estendere il gioco a vari giocatori con un campo più grande, introdurre un mostro che cercherà di mangiare uno dei giocatori, predisporre un programma che permetterebbe al Videopac di giocare, ecc.

```
00010 PRINT"Il Gioco di WYTTHOF 013@&"
00015 PRINT
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 X=3*INT(RND(1)*5)+5
00050 Y=INT(RND(1)*5)
00060 IF3*Y+5=XGOTO 50
00070 INIT 7:PAGE:STORE
00080 PRINT"   0 1 2 3 4 5 6 7 8 9"
00090 FOR I=0 TO 9
00100 PRINTI;" . . . . . . . . . ."
00110 NEXT I
00120 H=1:SCREEN:DISPLAY
00130 CURSORX X:CURSORY Y+3:PRINT"+"
00140 IFX>=22ANDY>=9THENSOUND 2:END
00150 CURSORX1:CURSORY15
00155 PRINT"GIOCATORE ";H
00160 INPUT "- Direzione (1,2,3)";D$
00170 D=VAL(D$)
00180 ONDGOSUB 300,400,500
00190 H=2-ABS(H-1)
00200 GOTO 130
00300 X=X+2:RETURN
00400 GOSUB 300
00500 Y=Y+1:RETURN
```

```
   0 1 2 3 4 5 6 7 8 9
0 . . . . . . . . . .
1 . . . . . . . . . .
2 . . . . . + + + . .
3 . . . . . . . + . .
4 . . . . . . . + . .
5 . . . . . . . + . .
6 . . . . . . . + . .
7 . . . . . . . + . .
8 . . . . . . . . + .
9 . . . . . . . . . +

GIOCATORE  1
- Direzione (1,2,3)? 2
```

Rimbalzi

RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: gioco di abilità.
Difficoltà del gioco: media.

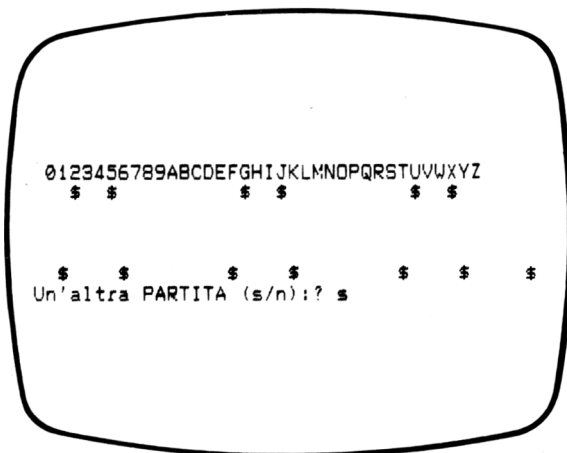
IL GIOCO: Utilizza l'idea della simulazione di un biliardo. Alle due estremità dello schermo si trovano la palla ed il bersaglio: una serie di cifre e di lettere, in alto sullo schermo, permette di scegliere il posto dove si desidera rimbalzare. La palla partirà allora verso l'alto (verso il basso se si introduce il segno —), rimbalzerà, ritornerà eventualmente in basso, rimbalzerà di nuovo, e così via fino a raggiungere la linea del bersaglio; se si è sufficentemente vicini al bersaglio si vince, ed il numero di rimbalzi sarà allora visualizzato.

IL PROGRAMMA: Le linee 20 e 50 stampano le cifre e le lettere in alto sullo schermo.
Le linee 60 e 70 definiscono le posizioni iniziali della palla e del bersaglio.
Le linee 80 e 90 effettuano la loro stampa.
Il simbolo corrispondente alla direzione scelta è inserito alla linea 90.
Il ciclo J (130÷170) realizza lo spostamento della palla.
La linea 160 calcola la nuova posizione.
Il cambiamento di direzione, durante lo scontro con un bordo, è effettuato dalla linea 160, per rinvio in una parte indipendente del programma, situata alle linee da 220 a 240.
La linea 250 effettua la stampa della fine della partita.
Il numero di rimbalzi è contato dalla linea 230.

PER GIOCARE: Si lancia il programma con RUN. Il campo appare, così come le posizioni della palla e del bersaglio. Si sceglie il posto dove si desidera far rimbalzare la palla (da 1 a Z), e lo si inserisce, seguito da RETURN.
La traiettoria è allora mostrata, seguita, se lo si merita, dal messaggio BRAVO; X RIMBALZI. Si può cominciare un'altra partita battendo un tasto qualunque, e fermarsi in ogni momento battendo RUN/STOP e RESTORE.

ESTENSIONI POSSIBILI: Sarà molto interessante, su un apparecchio che disponga di una grafica più fine, disegnare un vero biliardo, col suo tappeto verde, ed indicare la direzione con un segmento, che sim-

boleggi la coda del biliardo, che si sposta intorno alla biglia.
Potremmo anche aggiungere dei buchi, mettere varie biglie, sonorizzare i rimbalzi, o gli urti con altre palle, contabilizzarle per fare una partita con vari giocatori, ecc.

```
00010 PRINT" -RIMBALZI-  014@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 PRINT" 0123456789ABCDEFGHIJKLMN";
00055 PRINT"OPQRSTUVWXYZ"
00060 X=INT(RND(1)*18+2)
00070 Y=INT(RND(1)*18+2)
00080 CURSORX 1:CURSORY X:PRINT"$"
00090 CURSORX 39:CURSORY Y:PRINT"+"
00100 CURSORX 1:CURSORY 21
00105 INPUT "Tuo TIRO :";R$
00110 S=0
00120 T=X
00130 FOR J=1 TO 39
00140 D=ASC(R$)-ASC("0")+7*(R$>="A")
00145 T=T-X/D
00150 IFT>=21ORT<=1THEN220
00160 CURSORX J:CURSORY T:PRINT"$"
00170 NEXT J
00180 IFABS(T-Y)<3THENGOSUB 250
00190 INPUT"Un'altra PARTITA (s/n):";R$
00200 IFR$="n"THENEND
00210 INIT 7:RUN
00220 X=-X
00230 S=S+1
00240 GOTO 170
00250 PRINT"BRAVO :";S;" Rimbalzi"
00260 SOUND 2:RETURN
```

Jackpot

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: fortuna (un giocatore).
Difficoltà del gioco: facile.

IL GIOCO:

È una variante semplificata di un gioco di fortuna classico, nel quale il giocatore deve fare una scommessa, e tirare una leva per lanciare il gioco. Se ottiene certe combinazioni di simboli, vince (più o meno, secondo la sua puntata e la combinazione ottenuta). Qui, i simboli sono sostituiti da cifre, e il fatto di avere tre cifre identiche permette di vincere.
Esiste più di una combinazione vincente, che permette di avere il JACKPOT.

IL PROGRAMMA:

Il ciclo I (60÷75) permette il disegno della macchina, le linee 80÷95 visualizzano le locazioni (vuote) delle cifre.
La scommessa sarà inserita alla linea 105, e testata alla linea 110: se si scommette 0, la partita è terminata (partenza verso la linea 500).
Le linee 120, 140 e 160 permettono l'uscita delle tre cifre, richiamando un sottoprogramma che si trova alla linea 400.
Il risultato è visualizzato alle linee 165÷175, poi testato alla linea 180, e alla linea 310, se si vince.

PER GIOCARE:

Lanciare il programma con RUN; alla domanda CAPITALE INIZIALE?, si risponde inserendo un numero sufficientemente alto, sul quale le vostre puntate saranno decontabilizzate. Poi si batte RETURN, e si inserisce una prima scommessa (differente da 0), seguita da RETURN. Le tre cifre estratte sono allora visualizzate un breve istante, così come la puntata fatta, ed il risultato ottenuto (vincita o perdita).
Si può in seguito scommettere di nuovo, fare RETURN e così via. Per terminare, basta rispondere 0 alla domanda LA TUA PUNTATA.
Il resto del vostro capitale iniziale allora apparirà.

ESTENSIONI POSSIBILI:

Potremmo migliorare seriamente i controlli, usare i colori e delle possibilità grafiche più elaborate per simulare meglio un vero JACKPOT.

```
00010 PRINT"Jackpot=PENTOLACCIA 015@&"
00020 INPUT " Dammi un INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 PRINT:INPUT"MONTE di Partenza:";P
00050 INIT 7
00060 FOR I=1 TO 3
00065 CURSORX 10:CURSORY I+4:TX3,0,1
00070 PRINT"                      ":TX0,0,0
00075 NEXT I
00080 CURSORX 14:CURSORY 6:PRINT"   "
00085 CURSORX 19:CURSORY 7:PRINT"   "
00090 CURSORX 24:CURSORY 8:PRINT"   "
00095 CURSORX 19:CURSORY 9:PRINT"   "
00100 CURSORX 1:CURSORY 11
00105 INPUT "La tua PUNTATA :";M
00110 IFM=0GOTO 500
00120 GOSUB 400
00130 A=X
00140 GOSUB 400
00150 B=X
00160 GOSUB 400
00165 CURSORX 15:CURSORY 6:PRINTA
00170 CURSORX 20:CURSORY 6:PRINTB
00175 CURSORX 25:CURSORY 6:PRINTX
00180 IFA=BANDA=XGOTO 300
00190 N=-M
00200 CURSORX1:CURSORY11:PRINT" Punta";
00205 PRINT"ta:";M;" - Risultato:";N;"
00210 PRINT:INPUT"Con-RET-Continui";R$
00220 P=P+N
00230 GOTO 50
00300 N=M*50
00310 IFA=TTHENN=M*500
00320 GOTO 200
00400 X=INT(RND(1)*9+1):RETURN
00500 PRINT"Hai ";P;"  A Presto !?"
```

```
              5    6    5

 Puntata: 50 - Risultato:-50

Con-RET-Continui?
```

Roulette

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: fortuna (vari giocatori).
Difficoltà del gioco: facile.

IL GIOCO:

È soprattutto spettacolare per il suo aspetto grafico: una pallina, raffigurata da un quadrato nero, si sposta sulle caselle numerate da 1 a 36, così come le tre caselle supplementari, la 00, una casella P (come pari) e una casella D (come dispari).
La pallina percorre almeno una volta tutte le caselle, poi si ferma al secondo giro; si può in seguito rilanciarla per un'altra partita.

IL PROGRAMMA:

Il ciclo I (50÷110) disegna il tavolo di gioco, al quale la linea 80 aggiunge i tre simboli P, 00, e D.
Il ciclo I (130÷200) realizza lo spostamento del ciclo, l'arresto essendo controllato dalle linee 160÷180.
La linea 130 realizza un cancellamento particolare nel caso in cui la pallina passasse sui simboli P, 00, D.
Il sottoprogramma 300÷310 realizza una breve temporizzazione.
La linea 210 sceglie la posizione di arresto.
La linea 230 gestisce le giocate successive.

PER GIOCARE:

Basta battere RUN, e guardare. Una volta che la pallina si sia fermata, basterà battere un tasto qualunque (eccetto RUN/STOP) per rilanciarla. Battere RUN/STOP per uscire dal programma.

ESTENSIONI POSSIBILI:

Potremo cercare di avvicinarci maggiormente ad una vera roulette, in diversi modi: disegnare meglio il tavolo, introdurre i colori, simulare il rallentamento della pallina, introdurre differenti possibilità di scommessa, e assicurare la gestione delle vincite corrispondenti, tenere i conti di vari giocatori, ed annunciare i risultati in una maniera più usuale.

```
00010 PRINT"ROULETTE 016@&"
00020 INPUT "Dammi un INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE:STORE
00050 FOR K=1 TO 4
00060 FOR I=1 TO 9
00070 S=I+I
00075 Z=(SANDK=1)+(S+18ANDK=2)
00080 Z=Z+(S-1ANDK=3)+(S+17ANDK=4)
00085 PRINTTAB(4*I)Z;
00090 NEXT I
00100 PRINT:PRINT:PRINT
00110 NEXT K:SCREEN:DISPLAY
00115 CURSORX 1:CURSORY 3:PRINT"P"
00120 CURSORX 1:CURSORY 6:PRINT"00"
00125 CURSORX 1:CURSORY 9:PRINT"D"
00130 FOR K=1 TO 4
00140 FOR I=1 TO 10
00145 IFI=10ANDK<4THEN325
00150 CURSORX 4*I+2:CURSORY 3*K-2
00160 X=X-1:SOUND 4
00170 GOSUB 300
00180 IFX=1THENPRINT"*":GOTO 230
00190 NEXT I
00200 NEXT K
00210 X=INT(RND(1)*38)
00220 GOTO 120
00230 CURSORX 1:CURSORY 18
00235 INPUT "Un'altra Giocata(s/n):";R$
00240 IFR$="n"THENPRINT"GRAZIE !?":END
00250 I=10:K=4:NEXT I:NEXT K:GOTO 40
00300 FOR J=1 TO 15
00310 NEXT J:RETURN
00325 CURSORX 1:CURSORY 3*K-1:GOTO 160
```

```
     2   4   6   8   10  12  14  16  18
P

     20  22  24  26  28  30  32  34  36
00
             *
     1   3   5   7   9   11  13  15  17
D

     19  21  23  25  27  29  31  33  35




Un'altra Giocata(s/n):? s
```

Corridoi

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessione.
Difficoltà del gioco: difficile.

IL GIOCO:

Fa richiamo a delle nozioni di logica elementare. Il giocatore può presentarsi all'ingresso sia del corridoio n 1 che del n 2. In questi due ingressi sono piazzati degli sportelli che si spostano al vostro passaggio e vi lasciano passare, una volta a destra, una volta a sinistra. In più uno sportello analogo è stato messo nel corridoio centrale; potrete prevedere dove (uscita A o B) si effettuerà la vostra prossima uscita?

IL PROGRAMMA:

Le linee da 30 a 70 disegnano i corridoi (e i tre sportelli).
La linea 105 aspetta la vostra scelta.
La decodifica della strada è fatta dal Videopac utilizzando tre variabili logiche I, J, K, che rappresentano lo stato dei tre sportelli (linee da 130 a 270).

PER GIOCARE:

Dopo aver battuto RUN, il corridoio è disegnato sullo schermo, gli sportelli essendo visibili, e bisognerà battere 1 o 2 in risposta alla domanda del Videopac.
Nella versione che presentiamo la risposta è allora immediatamente visualizzata, e il percorso seguito dall'inizio del gioco, così come le uscite corrispondenti, appare sullo schermo. Ciò presenta il vantaggio di permettere a vari giocatori di esercitare la loro sagacità cercando di prevedere, per un certo ingresso, ciò che risponderà il Videopac.

ESTENSIONI POSSIBILI:

Sono molto numerose; è molto facile immaginare un corridoio più lungo e complesso, con altri ingressi ed altre uscite, e altri sportelli. Possiamo immaginare una visualizzazione grafica dei risultati disegnando, per un breve istante, lo stato degli sportelli dopo la risposta del giocatore. Possiamo anche aggiungere un test delle risposte (gioco in tanti), contabilizzare i punti dei giocatori, ecc.

```
00010 PRINT"CORRIDOI 017@&"
00020 INIT 7
00030 PRINT"---------------------"
00040 PRINT"B       ----------  X  2"
00050 PRINT"---------- X       ----"
00060 PRINT"A       ----------  X  1"
00070 PRINT"---------------------"
00080 I=0:J=0:K=0
00090 R$="":T$=""
00100 CURSORX 1:CURSORY 13
00105 INPUT"INGRESSO  1 o 2 :";C$
00110 IFC$=""ORC$>="3"GOTO 100
00115 IFLEN(C$)<>1GOTO 100
00120 R$=R$+C$
00130 PRINT:PRINTR$
00140 IFC$="1"GOTO 190
00150 J=NOTJ
00160 IFJ<>0 GOTO 250
00170 S$="B"
00180 GOTO 220
00190 I=NOTI
00200 IFI<>0 GOTO 250
00210 S$="A"
00220 T$=T$+S$
00230 CURSORX1:CURSORY17:PRINT"USCITA";
00235 PRINT" in ";S$:PRINT:PRINTT$
00240 GOTO 100
00250 K=NOTK
00260 IFK<>0GOTO 170
00270 GOTO 210
```

```
---------------------
B       ----------  X  2
---------- X       ----
A       ----------  X  1
---------------------






INGRESSO  1 o 2 :? 2

2

USCITA in B

B
```

Scacchiera

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: grafica.
Difficoltà del gioco: variabile, secondo l'utilizzo.

IL GIOCO:

Il programma proposto viene in appoggio al programma SCACCHI per essere utilizzato con una eventuale espansione di memoria; esso dà, infatti, la possibilità di realizzare una visualizzazione grafica di buona utilità per i giochi che utilizzano una scacchiera di 8*8 caselle (potremo facilmente estenderla a 10*10).
Una localizzazione è stata prevista, orizzontalmente, con delle lettere (da A ad H) e, verticalmente, con dei numeri (da 1 a 8).

IL PROGRAMMA:

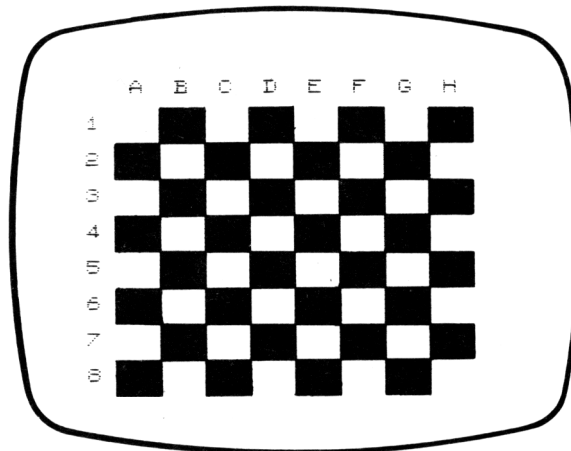
La linea 30 definisce una variabile O$ che contiene gli elementi del disegno. La linea 40 stampa le lettere orizzontali e la prima linea del disegno.
Il ciclo I (60÷150) permette la stampa, il carattere da stampare essendo visualizzato o no secondo il valore di (I+J)/2; le linee (100÷130) realizzano la stampa.

PER GIOCARE:

L'ordine RUN provoca l'apparizione della scacchiera sullo schermo.
Per utilizzare il programma, converrà tener conto della localizzazione delle caselle utilizzabili che sono i centri dei quadrati neri o grigi. Le caselle sono ottenute in larghezza da una tabulazione del tipo 3*I+1, I variante da 1 a 8, e in altezza da 2*I+1.

ESTENSIONI POSSIBILI:

Potremo usare questo programma per tutta una serie di giochi che si svolgano su una scacchiera (scacchi, dama, il lupo e l'agnello, ecc.).

```
00010 PRINT"SCACCHIERA 0180&"
00020 INIT 7
00030 O$=CHR$(127)+CHR$(127)+CHR$(127)
00040 PRINT"     A  B  C  D  E  F  G  H"
00050 PRINT
00060 FOR I=1 TO 8
00070 PRINT:PRINTI
00080 FOR J=1 TO 8
00090 IF(I+J)/2=INT((I+J)/2)GOTO 140
00100 CURSORX 3*J+1:CURSORY 2*I+2
00110 PRINTO$
00120 CURSORX 3*J+1:CURSORY 2*I+3
00130 PRINTO$
00140 NEXT J
00150 NEXT I
```

Il gioco della fava

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: media.

IL GIOCO:

Presenta una certa analogia con il gioco di WYTTHOF; questa versione offre infatti 4 possibilità di spostamento; la posizione di partenza è qui fissata in (0,0), e potrebbe eventualmente essere modificata.
Le direzioni di spostamento disponibili sono: destra, basso, diagonale in basso a destra, diagonale in basso a sinistra.
Si gioca in teoria con la regola: il primo arrivato in (9,9) ha perso.

IL PROGRAMMA:

La linea 30 ed il ciclo I (40÷60) disegnano il campo di gioco.
La linea 100 realizza il testo di fine partita.
La linea 130 permette di eseguire uno dei sottoprogrammi di spostamento.
Le linee 140 e 150 assicurano il cambiamento di giocatore.

PER GIOCARE:

Si lancia il programma con RUN, ed il campo di gioco appare, così come la domanda DIREZIONE (1, 2, 3, 4); le 4 direzioni proposte corrispondono agli spostamenti seguenti:

```
        1
4   3   2
```

Il giocatore 1 inserisce il suo spostamento, e batte RETURN.
Il movimento corrispondente è realizzato, poi si ottiene il messaggio GIOCATORE 2 e DIREZIONE... Il giocatore 2 procede nello stesso modo, e così via, fino a che uno dei giocatori arriva in (9,9) nel qual caso il gioco termina.

ESTENSIONI POSSIBILI:

Potremmo introdurre all'inizio della partita delle caselle con delle trappole (ognuno cercherà allora di fare finire l'altro in una di quelle caselle).
Si può anche giocare con vari simboli, se possibile di colore diverso, spostati simultaneamente, proporre dei campi di gioco triangolari, di dimensione arbitraria, ecc.

```
00010 PRINT"Il Gioco della FAVA 019@&"
00020 INIT 7:PRINT
00030 PRINT"   0 1 2 3 4 5 6 7 8 9"
00040 FOR I=1 TO 9
00050 PRINTI;" . . . . . . . . . .
00060 NEXT I
00070 X=4:Y=2
00080 J=1
00090 CURSORX X:CURSORY 2*Y:PRINT""
00100 IFX=22ANDY=6THENPRINT:GOTO 170
00105 CURSORX 1:CURSORY 18:
00110 PRINT"Giocatore ";J;
00115 INPUT " - Direzione(1,2,3,4)";I$
00120 I=VAL(I$):SOUND 2
00130 ONIGOSUB 200,300,400,500
00140 J=J+1
00150 IFJ=3GOTO 80
00160 GOTO 90
00170 PRINT"Il Giocatore";J;" ha Vinto"
00180 SOUND 2:SOUND 0:SOUND 5:SOUND 4
00190 PRINT"FINE";END
00200 Y=Y+1:RETURN
00300 GOSUB 200
00400 X=X+2:RETURN
00500 GOSUB 400:Y=Y-1:RETURN
```

```
   0 1 2 3 4 5 6 7 8 9
 1     . . . . . . . .
 2 . . . . . . . . . .
 3 . .   . . . . . . .
 4 . . . . . . . . . .
 5 . .   .   . . . . .
 6 . . . . . . . . . .
 7 . . .     . .   .
 8 . . . . . . . . . .
 9 . . . . .

Il Giocatore 2 ha Vinto


Giocatore  1 - Direzione(1,2,3,4)? 1
```

Gioco dell'oca aleatorio

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: fortuna (due giocatori).
Difficoltà del gioco: facile.

IL GIOCO:

Il gioco riprende l'antichissima idea del gioco dell'oca, ma aggiungendoci la possibilità offerta dal calcolatore di scegliere a caso le caselle "con problema". Le regole del gioco non hanno molto interesse, visto che i giocatori non intervengono in nessun momento nello svolgimento della partita, che è interamente gestita dal Videopac; basta schiacciare uno dei tasti dopo il lancio dei dadi, e guardare cosa succede.

IL PROGRAMMA:

La linea 60 permette di chiamare il sottoprogramma (1000-1180) che visualizza il tavolo di gioco, il controllo della posizione dei giocatori e la visualizzazione di quest'ultima. Questa visualizzazione sarà d'altronde rifatta durante la partita, per tener conto dello spostamento realmente effettuato (che non è necessariamente quello indicato dal dado). La linea 70 si occupa del cambiamento del giocatore, e la linea 80 effettua il lancio del dado. La linea 85 si occupa del caso in cui uno dei giocatori termina. La linea 130 prende un numero aleatorio tra 0 e 3, che servirà, se sarà il caso, a determinare il vero spostamento del giocatore, rinviando in uno dei sottoprogrammi di "fortuna".
Le linee 1000 e 1010 fanno avanzare le pedine, e le linee 1020 e 1080 ridisegnano il tavolo di gioco ad ogni turno.

PER GIOCARE:

Si lancia il gioco con RUN. Il tavolo di gioco è allora disegnato, poi appare il messaggio GIOCATORE 1:——. Si dovrà allora battere su un tasto qualsiasi perché il colpo del giocatore 1 sia preso in considerazione; non preoccupatevi per il tempo di reazione, che potrà sembrare un po' lungo.
Lo spostamento indicato dai dadi sarà eseguito, poi, dopo qualche istante, un secondo spostamento sarà evenualmente fatto: è lo spostamento "fortuna". Apparirà allora il messaggio GIOCATORE 2:——. Il giocatore 2 batte a sua volta un tasto qualsiasi e così via...
Il giocatore vincente sarà segnalato da un messaggio BRAVO in fine di partita.
Si può fermare il gioco in ogni momento battendo RESTORE.

ESTENSIONI POSSIBILI: Si potrebbe ingrandire il tavolo di gioco, introdurre un maggior numero di messaggi aleatori, cambiare la loro natura. Potremmo anche scegliere le caselle "fortunate" all'inizio del gioco, aggiungere dei messaggi di conseguenza, ecc.

```
00010 PRINT"Gioco dell'OCA 020@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 X=0:Y=0:J=1:I=X
00060 GOSUB 1000
00070 J=ABS(J-1)
00080 I=INT(RND(1)*6+1)
00085 IFX<>33ORY<>35GOTO 100
00090 CURSORX1:CURSORY20:PRINT"BRAVO !"
00095 SOUND 3:SOUND 4:SOUND 6:END
00100 SOUND 0:CURSORX 1:CURSORY 12
00105 PRINT"Giocatore ";J+1;
00110 PRINT" - tuo COLPO :";I
00115 PRINT:INPUT "Premere -RET- ";R$
00120 GOSUB 1000
00130 I=INT(RND(1)*4)
00140 F=INT(RND(1)*5+1)
00150 ONFGOSUB 200,300,400,500,160
00160 GOTO 60
00200 I=I+3:RETURN
00300 I=-I:RETURN
00400 I=I*2:RETURN
00500 IFJ=0THENX=INT(RND(1)*30):RETURN
00510 Y=INT(RND(1)*30):RETURN
01000 IFJ=0THENX=X-I*(X+I<=35)
01010 Y=Y-J*I*(Y+I<=35)
01020 CURSORX 1:CURSORY 3:TX0,4,0
01025 PRINT"01213456789ABCDEFGHIJKLMN";
01030 PRINT"OPQRSTUVWXYZ":TX0,0,0
01035 FOR I=1 TO 35
01040 CURSORX I:CURSORY 5:PRINT" ";
01050 IFI=XTHENGOSUB 1150
01060 CURSORX I:CURSORY 6:PRINT" ";
01070 IFI=YTHENGOSUB 1170
01080 NEXT I
01090 FOR I=1 TO 600:NEXT I
01100 RETURN
01150 CURSORXI:CURSORY5:PRINT"1";
01160 RETURN
01170 CURSORXI:CURSORY6:PRINT"2";
01180 RETURN
```

```
01213456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

1




Giocatore  1 - tuo COLPO : 1

Premere -RET- ?
```

LIVELLO 2

N.	Nome del programma	Tema e particolarità
21	Dado truccato	Da utilizzare in altri programmi, se siete disonesti.
22	Visioni	L'elaboratore gioca sulle parole.
23	4 2 1	Simulazione semplificata, per più giocatori.
24	L'elettricista matto	Difficile gioco di logica, per un solo giocatore.
25	La corsa al 20	Gioco di riflessione per due o più giocatori.
26	Musica	Ripassate le vostre nozioni di solfeggio con l'elaboratore.
27	Poker	Base di partenza per la messa a punto di vostri programmi.
28	Cavallina	Gioco di riflessione, per due giocatori.
29	Lettere	Semplicissimo gioco di parole.
30	Macina-parole	L'elaboratore crea nuove parole.
31	Spia	Fatele una dichiarazione in codice.
32	Agganciate i vagoni	Molto visivo, per i più giovani.
33	Lancio d'un dado truccato	Ripassatevi le nozioni di statistica
34	Giro di carte	L'elaboratore è pronto a smazzare.
35	Il gioco di Marienbad	Gioco di logica per due giocatori, in versione estesa.
36	L'indesiderabile	Quale proprietario, vi riuscirà d'acciuffare l'inquilino abusivo?
37	Componi	Per far musica occorre scriverla.
38	Giorno della settimana	Approfittatene per fare il punto sugli anniversari.
39-40	Il gioco della vita	Un gran classico, in due versioni: manuale ed automatica.

Dimensionamento di una tabella Ricerca di un elemento in una tabella e in una stringa

L'ISTRUZIONE DIM, O ISTRUZIONE DI DIMENSIONAMENTO DI UNA TABELLA

Le strutture di una tabella sono molto correntemente utilizzate in programmazione, si rivelano anzi indispensabili in un gran numero di casi. Se volete, per esempio, inserire le vostre entrate e uscite per tutti i giorni del mese (supporremo, per semplificare, dei mesi di trenta giorni) potreste definire 30 variabili chiamate PRIMO GIORNO, SECONDO GIORNO, ecc. È più facile, e più razionale, utilizzare un ciclo I, I variante da 1 a 30, e una variabile E(I) per le entrate, e U(I) per le uscite; così facendo, avete, senza saperlo, usato una struttura di tabella a una dimensione; questo sarà ammesso dal VIDEOPAC+ solo se lo avete avvertito prima che avreste usato una simile tabella, dando un'istruzione DIM E(30) per le entrate, e DIM U(30) per le uscite.
Potremmo anche conservare le entrate e le uscite in una stessa tabella, dandogli questa volta due dimensioni: per esempio DIM EU(30,2) (o DIM EU(2,30)). Così, con DIM (30,2) si converrà che EU(I,1) rappresenta le entrate, e EU(I,2) le uscite. Delle tabelle con più di due dimensioni sono ugualmente possibili.
Si possono anche concepire delle tabelle i cui elementi sono delle stringhe di caratteri, invece che dei numeri, con 255 caratteri al massimo.
Così DIM NOME$ (12) riserva 12 spazi in memoria, per delle parole di 255 caratteri al massimo.
Sul VIDEOPAC+, non è obbligatorio dimensionare una tabella con meno di 10 elementi.

RICERCA DI UN ELEMENTO IN UNA TABELLA: R(I)

Se si ha una tabella a una dimensione, numerico o alfanumerico, definito da un'istruzione come DIM R(30) o DIM NOME$ (20), il primo elemento di questa tabella è localizzato da R(I) (o N$ (I)); sarà allora manipolato esattamente come una variabile ordinaria (assegnazione, calcolo, tests, ecc.).
Così INPUT R(3) permette di inserire il valore del terzo elemento numerico della tabella R. PRINT R$ (5) permette la visualizzazione del quinto elemento della tabella alfanumerica R$, ecc.

RICERCA DI UN ELEMENTO IN UNA STRINGA: MID$

Per estrarre l'i-esimo elemento della stringa alfanumerica A$, si utilizzerà l'istruzione MID$.
Così, se A$="BUONGIORNO", MID$ (A$,3,1) darà il carattere O; inoltre MID$ (A$,3,3) darà la sottostringa "ONG".

TRATTAMENTO DI DATI MULTIPLI, ISTRUZIONI READ, DATA, RESTORE

Per riempire facilmente una tabella, numerico o alfanumerico, sarà interessante utilizzare delle linee DATA, che contengono i dati scritti gli uni dopo gli altri, separati da una virgola.
Così, potremo scrivere:

DATA DALIA, VIA SAN VITTORE 45, SERGIO...

Questi dati possono essere letti nei cicli FOR———NEXT, tramite l'istruzione READ. Si può ricominciare la lettura all'inizio delle DATA tramite l'istruzione RESTORE.

Dado truccato

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: facile.

IL GIOCO:

Si vuole cercare di fabbricare un dado truccato secondo la propria convenienza. Il metodo proposto qui consiste nel definire all'inizio una probabilità di realizzazione per ogni tipo di lancio (cioè la probabilità di ottenere un 1, un 2, un 3, ecc. che ognuna di queste probabilità sia compresa tra 0 e 1, e che la somma di tutte le probabilità sia un numero molto vicino a 1).

IL PROGRAMMA:

Utilizza il generatore aleatorio del Videopac supposto di qualità adeguata (potremo facilmente usare un qualunque altro generatore se ce ne sarà bisogno).
Il ciclo I (70÷140) si occupa dell'ingresso di ognuna delle probabilità, con verifica ad ogni ingresso e sul totale (linea 110); ogni probabilità è conservata in una variabile indicizzata E(I): linea 95 del programma.
Il lancio del dado truccato è effettuato con l'aiuto di un altro ciclo I (160÷200), la linea che permette il trucco è la 190.
Un nuovo lancio potrà essere effettuato con l'uso del tasto RET (linea 220).

PER GIOCARE:

Dopo la partenza con la battuta di RUN, si ottiene la domanda PROBABILITÀ DEL DADO 1, alla quale si dovrà rispondere con un numero compreso tra 0 e 1. Converrà ricordarsi che il totale di tutte le probabilità inserite non dovrà in alcun caso superare 1 (tolleranza di 0.1).
Dei messaggi di errore potranno apparire ed il programma sarà eventualmente rilanciato.
Quando tutte le probabilità saranno state definite, il primo lancio sarà automaticamente effettuato, e basterà premere il tasto RET per effettuare i lanci successivi. Si arresta il programma battendo RESTORE.

ESTENSIONI POSSIBILI:

Ci sarebbe evidentemente la forte tentazione di utilizzare subdolamente questo piccolo programma per aumentare le vostre possibilità di vittoria in uno dei giochi di fortuna che usano il lancio di dadi. Si potrebbe allora pensare di inserire i dati che definiscono il dado truccato in una variabile A$, ecc. Potremmo anche im-

maginare di presentare i risultati corrispondenti ad una serie di lanci in una maniera spettacolare, che metta il trucco in evidenza...

```
00010 PRINT"DADO Truccato 021@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 P=0
00060 DIM E(6)
00070 FOR I=1 TO 6
00080 CURSORX 17:CURSORY 3
00085 PRINT"                          "
00090 CURSORX 1:CURSORY 3
00095 PRINT"Probabilita' del DADO ";I;
00100 INPUT E(I)
00105 P=P+E(I)
00110 IFE(I)<0ORE(I)>1ORP>1.1GOTO 240
00120 CURSORY6:PRINTTAB(6*I-5):"DADO":I
00130 PRINT:PRINTTAB(6*I-5);"Prob":E(I)
00140 NEXT I
00150 IFP<.9THENSOUND 4:RUN40
00160 FOR I=1 TO 500:NEXT I
00170 N=INT(RND(1)*6+1)
00180 FOR I=1 TO 6
00190 IFN=1ANDRND(1)>E(I)THEN290
00200 NEXT I
00210 CURSORX1:CURSORY15:PRINT"TIRO :";
00215 PRINTN:"            ":SOUND 4:PRINT
00220 INPUT"Con RET Continui i TIRI";R$
00230 GOTO 170
00240 CURSORX 1:CURSORY 4
00245 PRINT"IMPOSSIBILE :0<= P <=1"
00250 P=P-E(I)
00260 FOR J=1 TO 1500:NEXT J
00270 CURSORX 1:CURSORY 4:PRINT"       ";
00275 PRINT"                      "
00280 GOTO 90
00290 I=6:NEXT I:GOTO 170
```

Visioni

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: disegno.
Difficoltà del gioco: facile.

IL GIOCO:

Il Videopac vi permette, per la prima volta, di entrare nella sua intimità lasciandovi intravedere la sua propria rappresentazione del mondo! Dovrete infatti proporgli delle parole, non importa quali, e lui vi risponderà proponendovi una raffigurazione grafica di ciò che rappresenta quella parola per lui.
Il gioco sarà molto più divertente se lo farete in molti, e potrete a volte avere delle sorprese.

IL PROGRAMMA:

Un ciclo I (40-120) si incarica di realizzare i disegni.
La linea 50 permette di trasformare in numero ogni lettera della parola inserita all'inizio (linea 30).
Questo numero sarà in seguito portato in base 2 dal ciclo J (linee 60-100); secondo il risultato di questa trasformazione, un punto sarà stampato, oppure no (linea 90).

PER GIOCARE:

Si inizia l'esecuzione del programma con RUN, il che porta all'apparizione della domanda CHE DISEGNO? Si risponde inserendo una parola qualunque che sarà trasformata in disegno dal Videopac non appena avremo battuto RETURN. Per ricominciare basterà premere un tasto qualunque, mentre per finire si batte RESTORE.
Vi lascio gustare il disegno che realizzerà il Videopac quando avrà conosciuto il nome dell'autore di questo libro.

ESTENSIONI POSSIBILI:

La stessa idea può essere ripresa introducendo degli elementi di simmetria nei disegni, aggiungendoci i colori. Possiamo anche immaginare di codificare con questo metodo un certo numero di disegni, che introdurremo man mano nella macchina, e potremo allora creare un gioco nel quale il Videopac vi chiede di trovare la parola che corrisponde al disegno, ecc.

```
P
00010 PRINT"VISIONI ?!? 022@&"
00020 INIT 7
00030 INPUT "Che DISEGNO :";Z$
00040 FOR I=1 TO LEN(Z$)
00050 N=9*ASC(MID$(Z$,I,1))-200
00060 J=1
00070 S=N/2-INT(N/2)
00080 N=INT(N/2)
00090 IFS=0THENJ=J+1:GOTO 105
00095 CURSORX J+15:CURSORY I+5
00100 TX0,4,0:PRINT" ":TX0,0,0
00105 J=J+1
00110 IFJ<10GOTO 70
00120 NEXT I
00130 PRINT:PRINT
00140 INPUT "Batti-ENTER-";Z$
00150 GOTO 20
```

CHE DISEGNO:? JACQUES

BATTI – ENTER –

421

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: media.

IL GIOCO:

Ogni giocatore dovrà, nel suo turno, lanciare tre dadi, e può in seguito scegliere, sui due tiri successivi, se desidera o no rilanciare 1, 2 o 3 dadi. Ottiene allora un punteggio corrispondente al numero indicato dai dadi, secondo le regole scelte, ed il giocatore successivo procede nella stessa maniera.

IL PROGRAMMA:

Il numero di giocatori è innanzitutto inserito alla linea 40.
La linea 110 visualizza il numero del giocatore e il numero di tentativi già effettuati.
Un sottoprogramma posizionato alla linea 4000, utilizzato tre volte nel ciclo J (200÷250), realizza il lancio dei dadi.
I rilanci sono eseguiti dalle linee 1000, 2000, 3000.
La variabile R$ genera i secondi e terzi lanci.
La stampa del risultato sarà in tutti i casi realizzata dalle linee 140÷160.
I test per il numero di tentativi e per il cambio di giocatore sono effettuati alle linee 260 e 280.

PER GIOCARE:

Si lancia l'esecuzione del programma con RUN, il che porta al messaggio NUMERO DI GIOCATORI? Si inserisce questo numero, e si batte RETURN. Lo schermo si cancella e visualizza: GIOCATORE N. 1. PROVA A, PARTITA P, così come il risultato realizzato, e il numero dei dadi (A, B, C).
Se non si desidera rifare dei lanci, basta battere RETURN. La mano passa allora al giocatore successivo. Per effettuare un altro lancio, basta battere il nome del o dei dadi da rilanciare, e battere RETURN. I giocatori successivi procederanno nello stesso modo.

ESTENSIONI POSSIBILI:

La più evidente consiste nel lasciare la gestione dei risultati ai giocatori, riferendosi alle regole di vincita usuali del gioco 421. Potremmo fare apparire dopo ogni turno una tabella con la ricapitolazione dei risultati dei giocatori.
Potremmo anche migliorare la visualizzazione dei dadi (riferirsi ai vari programmi di visualizzazione dei dadi presenti nella stessa opera).

```
00010 PRINT" - 421 -  023@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INPUT "Numero di GIOCATORI :";N
00050 P=1
00060 I=1
00070 K=0
00080 INIT 7
00090 R$="ABC"
00100 GOTO 200
00110 K=K+1
00120 CURSORX 16:CURSORY 1:PRINT"-421-"
00125 TX0,0,0:CURSORX 1:CURSORY 4
00130 PRINT"Giocatore Num.";I;" PROVA";
00135 PRINT" ";K;" -Partita ";P
00140 CURSORX 7:CURSORY 7:PRINTA;
00145 PRINTSRC(10);B;SPC(10);C
00150 CURSORX 8:CURSORY 9:PRINT"A";
00160 PRINTSPC(11);"B";SPC(11);"C"
00165 R$="":CURSORX1:CURSORY14
00170 PRINT"                          ":CURSORY 14
00180 INPUT"ABC: ";R$:IFR$=""GOTO 270
00190 R$=R$+"DDD"
00200 FOR J=1 TO 3
00210 GOSUB 4000
00220 X$=MID$(R$,J,1)
00230 X=ASC(X$)-64
00240 ONXGOSUB 1000,2000,3000,4000
00250 NEXT J
00260 IFK<3GOTO 110
00270 I=I+1
00280 IFI<=NGOTO 70
00290 P=P+1
00300 GOTO 60
01000 A=D:RETURN
02000 B=D:RETURN
03000 C=D:RETURN
04000 D=INT(RND(1)*6+1):RETURN
```

```
                 -421-

Giocatore Num. 1 PROVA  1 -Partita  1

       1             6             5

       A             B             C



ABC: ? CAB
```

L'elettricista matto

RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: abbastanza difficile.

IL GIOCO: Un palazzo con 10 finestre, numerate da 0 a 9, appare sullo schermo. Il giocatore dovrà ad ogni tentativo manovrare uno degli interruttori, indicando il suo numero (da 0 a 9).
In teoria, la finestra corrispondente dovrebbe spegnersi, ma è anche possibile che un'altra si illumini (o si spenga). Si tratta, naturalmente, di cercare di ottenere lo spegnimento delle 10 finestre, col minor numero di tentativi possibile.

IL PROGRAMMA: Una tabella A, alla linea 50, conterrà lo stato (acceso o spento) delle 10 finestre.
Il ciclo I (70-90) e l'altro ciclo I (110-160) disegnano il palazzo nel suo stato iniziale.
La linea 170 controlla se tutto è spento, nel qual caso si andrà alla linea 310 per visualizzare il numero di tentativi.
Le linee 180-185 attendono l'indicazione dell'interruttore utilizzato.
Le linee da 190 a 290 definiscono il risultato della manovra.

PER GIOCARE: RUN provoca l'apparizione del palazzo, con tutte le finestre accese.
Alla domanda INTERRUTTORE NUM.? si risponde indicando il numero dell'interruttore da usare, seguito da RETURN.
Il risultato della manovra è allora indicato sullo schermo (delle finestre si accendono o si spengono). Si continua così fino a che tutte le finestre sono spente, e si ottiene allora il numero di tentativi fatti.

ESTENSIONI POSSIBILI: Potremmo immaginare una grafica molto migliore, con, se possibile, i colori. Potremmo anche fare un gioco con due giocatori, aumentare il numero di finestre, modificare la routine per l'accensione o lo spegnimento...

```
00010 PRINT"Elettricista MATTO 024@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 M=0
00050 DIM A(10)
00060 INIT 7
00070 FOR I=1 TO 7
00080 TX3,0,0:CURSORX 1:CURSORY I
00085 PRINT"                                "
00090 NEXT I
00100 TX0,0,0
00110 FOR I=1 TO 10
00120 T=T+A(I)
00130 CURSORX 2*I+5:CURSORY 3
00135 TX2,4,0:PRINT" "
00140 IFA(I)=0THENTX1,0,0
00145 CURSORX 2*I+5:CURSORY 3:PRINT" "
00150 TX0,0,0:CURSORX 2*I+4:CURSORY 6
00155 PRINTI-1
00160 NEXT I
00170 IFT=10GOTO 310
00180 CURSORX 1:CURSORY 14:SOUND 0
00185 INPUT "INTERUTTORE Num.:";N
00190 N=N+1
00200 M=M+1
00210 IFA(N)=1GOTO 290
00220 A(N)=1
00230 R=TAN(RND(1)+N/RND(1)-N)
00235 R=R-SIN(RND(1)/N)+336*SIN(8*N)
00240 N=INT(10*(R-INT(R)))
00250 IFN=0THENN=INT(RND(1)*10+1)
00260 IFA(N)=1GOTO 290
00270 A(N)=1
00280 GOTO 100
00290 A(N)=0
00300 GOTO 230
00310 PRINT:PRINT:"TROVATO in ";M;" Colpi"
```

```
  ■   ■   ■ ■   ■
0 1 2 3 4 5 6 7 8 9

INTERRUPTEUR NUMERO :? 2
```

La corsa al 20

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: logica (due o più giocatori).
Difficoltà del gioco: media.

IL GIOCO:

È possibile immaginare per questo gioco varie regole, e abbiamo preso qui solo la regola usuale per due giocatori, che ne fa un gioco simile al gioco dei fiammiferi, o ai giochi di spostamento: 4 mucchi di 20 stelle appariranno all'inizio della partita, ed ogni giocatore può prendere tante stelle quante ne desidera, ma questo in un solo mucchio per volta. Chi prende l'ultima stella è considerato perdente.

IL PROGRAMMA:

Il ciclo I (30÷50) disegna 4 file di 20 stelle.
La variabile F, dimensionata alla linea 70, servirà a registrare il numero di stelle prese in ogni fila.
La variabile T conterrà il numero totale di stelle prese.
La linea 120 controlla che ci siano abbastanza stelle sulla linea indicata.
I cicli I (150÷190) e J (160-180) levano le stelle; la linea 130 testa la fine della partita.

PER GIOCARE:

Lanciare il programma con RUN; il tavolo di gioco appare, poi i messaggi GIOCATORE 1 e FILA, poi STELLE? Si risponde inserendo un numero di fila (tra 1 e 4) seguito da RETURN, ed il numero di stelle da levare seguito da RETURN. Le stelle sono allora eliminate, e toccherà al secondo giocatore. La partita termina, quando l'ultima lettera è stata tolta, con la visualizzazione del giocatore vincente.

ESTENSIONI POSSIBILI:

Oltre ad una miglioria della qualità grafica e dei controlli, numerose varianti sono ipotizzabili, modificando se necessario le regole di partenza: possiamo autorizzare la presa solo di un numero dispari di stelle, o alternare pari e dispari, oppure prendere a caso il numero di stelle da levare, o ancora giocare in tanti...

```
00010 PRINT"La CORSA AL 20 - 025@&"
00020 INIT 7
00030 FOR I=1 TO 4
00040 CURSORX1: CURSORY 2*I:PRINTI;
00045 PRINT"   ********************"
00050 NEXT I
00060 T=0
00070 DIM F(4)
00080 B=1
00090 CURSORX 1:CURSORY 12
00095 PRINT"GIOCATORE Num. :";B
00100 PRINT:INPUT "LINEA :";L
00110 PRINT:INPUT "Num.CASELLE :";C
00120 IF(C+F(L))>20GOTO 90
00130 F(L)=F(L)+C
00140 T=T+C
00150 FOR I=1 TO 4
00160 FOR J=(F(I)-C) TO F(I)
00170 CURSORX 26-J:CURSORY 2*I
00175 PRINT" ":SOUND 0
00180 NEXT J
00190 NEXT I
00200 B=B+1
00210 IFB=3THENB=1
00220 IFT<80GOTO 90
00230 PRINT"Il Giocatore Num. :";B;
00240 PRINT" VINCE !!!":SOUND 2:SOUND 4
```

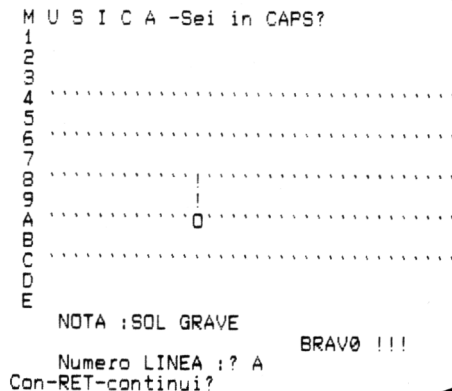
```
  1   ********************

  2   ********************

  3   *************

  4   ********************


GIOCATORE Num.   : 1

LINEA :? 3

Num.CASELLE :? 7
```

Musica

RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: gioco educativo.
Difficoltà del gioco: media.

IL GIOCO: Si presenta come un esercizio di insegnamento del solfeggio: l'apparecchio annuncia una nota (in chiave di sol), seguita dal messaggio GRAVE o ACUTA, secondo la sua posizione sullo spartito. Si dovrà rispondere indicando su quale linea la nota apparirà; la nota è allora disegnata sullo spartito, seguita da un messaggio BRAVO o PERSO, secondo che si abbia vinto o meno.

IL PROGRAMMA: Il ciclo I (50÷70) disegna lo spartito.
Un secondo ciclo I (80÷120) visualizza i numeri di linea.
La linea 130 definisce il nome delle note, e la linea 140 realizza la presa aleatoria di una nota.
Il nome della nota è calcolato alla linea 150 ed è visualizzato dalle linee (160÷165), seguito dal messaggio GRAVE o ACUTA, secondo il caso.
La risposta sarà inserita alle linee 190-200.
Le linee (200÷210) si occupano della decodifica della risposta.
Le linea (230÷250) visualizzano la nota.
Le linee 270 e 280 guardano se si ha vinto o perso e visualizzano il messaggio corrispondente.
Le linee da 290 a 300 permettono di fare una nuova prova.

PER GIOCARE: Si lancia il programma con RUN. Si ottiene allora l'apparizione dello spartito, così come il nome della nota da posizionare e la domanda Numero DI LINEA.
Si risponde dando un numero di linea (da 1 a E) seguito da RETURN. La nota è allora disegnata e il messaggio BRAVO o PERSO appare, in questo caso, sullo schermo.
Si deve battere RET per continuare, e ci si fermar battendo RESTORE.

ESTENSIONI POSSIBILI: Possiamo estendere ad un'altra chiave, che non sia la chiave di sol: possono introdurre il colore, i suoni, ecc.

```
00010 PRINT" *MUSICA*  026@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE:STORE:DELIM0,7,4
00045 PRINT" M U S I C A -Sei in CAPS?"
00050 FOR I=1 TO 5
00060 CURSORX 4:CURSORY 2*I+4
00065 FOR J=1TO 34:PRINTCHR$(96);:NEXTJ
00070 NEXT I
00080 FOR I=1 TO 14
00090 CURSORX 1:CURSORY I+2
00100 IFI<=9THENPRINTI
00110 IFI>9THENPRINT" ";CHR$(I+55)
00120 NEXT I:SCREEN:DISPLAY
00130 N$="SI LA SOLFA MI RE DO "
00140 X=INT(RND(1)*14+1)
00150 T=X-1-7*INT((X-1)/7)
00160 CURSORX5:CURSORY17:PRINT"NOTA :";
00165 PRINTMID$(N$,3*T+1,3);
00170 IFX<=7THENPRINT" ACUTO":PRINT
00180 IFX>7THENPRINT" GRAVE":PRINT
00190 PRINTTAB(4)"Numero LINEA :";
00200 INPUT L$:L=-(ASC(L$)-48)*(L$<":")
00210 L=L-(ASC(L$)-55)*(L$>="A")
00220 Z$=CHR$(125)
00230 CURSORX 16:CURSORY X:PRINTZ$
00240 CURSORX 16:CURSORY X+1:PRINTZ$
00250 CURSORX 16:CURSORY X+2:PRINT"O"
00260 CURSORX 25:CURSORY 18:TX0,4,0
00270 IFL=XTHENPRINT"BRAV0 !!!":PRINT
00280 IFL<>XTHENPRINT"PERSO !?!":PRINT
00290 TX0,0,1:PRINT"Con-RET-continui";
00300 TX0,0,0:INPUTR$:GOTO 40
```

```
 M U S I C A -Sei in CAPS?
 1
 2
 3
 4  ``````````````````````````````````
 5
 6  ``````````````````````````````````
 7
 8  ````````````!`````````````````````
 9              !
 A  ````````````O`````````````````````
 B
 C  ``````````````````````````````````
 D
 E
    NOTA :SOL GRAVE
                        BRAV0 !!!
    Numero LINEA :? A
Con-RET-continui?
```

Poker

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco contro il Videopac.
Difficoltà del gioco: facile.

IL GIOCO:

È una simulazione di una partita di POKER tradizionale, effettuata con un mazzo di 32 carte, nella quale il giocatore avrà, come avversario, il Videopac.
Il giocatore riceverà 5 carte, individuate con dei numeri da 1 a 5, e potrà decidere di cambiarne alcune annunciando quelle che scarta.
Le sue nuove carte saranno allora visualizzate, ed è in seguito il turno di gioco del Videopac.

IL PROGRAMMA:

Le linee 50 e 60 definiscono le carte.
Una variabile N è dimensionata a 16, alla linea 50, per ricevere 10 carte giocatore e 5 carte Videopac.
La presa sarà effettuata da un ciclo I (80÷130) ed il controllo da un ciclo J (110÷125).
La visualizzazione delle carte è realizzata dal sottoprogamma 500÷630.
Un altro ciclo I (190÷210) realizza la seconda presa del giocatore, la linea 240 permette la seconda visualizzazione, e il ciclo I (500-570), dove si entra direttamente dopo, realizzerà la visualizzazione del gioco del Videopac.

PER GIOCARE:

L'ordine RUN lancia l'esecuzione del programma, e le 5 prime carte del giocatore appaiono allora. Alla domanda CARTE Rifiutate? il giocatore può rispondere 0 seguito da RETURN se vuole fermare la partita. Per giocare, risponderà semplicemente RETURN se si stima servito, oppure inserirà i numeri delle carte che rifiuta seguiti da RETURN.
Per esempio, se batte 235 e RETURN, il suo nuovo gioco conserverà le carte 1 e 4. Le sue carte saranno allora visualizzate così come quelle del Videopac, quest'ultimo non avendo, in questo programma, la possibilità di cambiare delle carte.
Basterà in seguito battere RETURN per cominciare un'altra mano.

ESTENSIONI POSSIBILI:

Saranno, naturalmente, molto numerose, da una visualizzazione grafica a colori delle carte fino al miglioramento del gioco del Videopac, l'introduzione delle regole sulle scommesse ed anche, perché no, la possibilità di bluff.

```
00010 PRINT"-POKER- 027@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 DIM N(16)
00050 V$="7 8 9 10FADAREAS"
00060 C$="FiQuCuPi"
00070 INIT 7:PAGE
00080 FOR I=1 TO 16
00090 N(I)=INT(32*RND(1))
00100 IFI=1THEN130
00110 FOR J=1 TO I-1
00120 IFN(I)=N(J)THENJ=I-1:NEXTI:GOTO90
00125 NEXT J
00130 NEXT I
00140 K=0:P=0
00150 CURSORX 1:CURSORY 16
00160 PRINT"CARTE Rifiutate"
00170 GOSUB 500
00180 IFR$="0"THENPRINT"A Presto!?":END
00190 FOR I=1 TO LEN(R$)
00200 N(VAL(MID$(R$,I,1)))=N(I+5)
00210 NEXT I
00220 INIT 7
00230 P=1
00240 GOSUB 500
00250 K=K+10
00500 FOR I=1 TO 5
00510 J=2*(N(I+K)-4*INT(N(I+K)/4)+1)-1
00520 F=2*(N(I+K)-7*INT(N(I+K)/7)+1)-1
00530 R$=MID$(V$,F,2)
00540 CURSORX 7*I-4:CURSORY K+2:PRINTI
00550 CURSORX 7*I-5:CURSORY K+4
00555 PRINTR$;":";MID$(C$,J,2)
00560 CURSORX 7*I-5:CURSORY K+5
00570 PRINT"-----":NEXT I
00580 IFP=1THENP=0:RETURN
00590 IFK=10THENTX0,4,1:GOTO 620
00600 CURSORX 18:CURSORY 16:INPUT R$
00610 RETURN
00620 PRINT:INPUT"Con-RET-Continui";R$
00630 TX0,0,0:GOTO 70
```

```
   1      2      3      4      5

 10:Qu  10:Pi  9 :Qu  RE:Pi  7 :Pi
 -----  -----  -----  -----  -----



   1      2      3      4      5

 8 :Qu  DA:Qu  RE:Qu  10:Pi  7 :Pi
 -----  -----  -----  -----  -----

Con-RET-Continui?
```

Cavallina

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: facile.

IL GIOCO:

Anche se l'idea sia in sé molto semplice, non sarà obbligatoriamente evidente scoprire una strategia adeguata. Disponiamo di tre pecore nere e di tre pecore grigie, raffigurate con le lettere M e W; le pecore possono spostarsi da una casella ad una casella libera. Varie regole potranno intervenire, secondo la vostra stima personale dell'interesse per una o per l'altra: o si costringono le pecore ad andare sulla prima casella disponibile, saltando eventualmente sopra le pecore avversarie, oppure si lascia loro scegliere la casella di arrivo, il che può rendere le partite interminabili; a voi la scelta.

IL PROGRAMMA:

Il ciclo I (50-80) inizializza una tabella A (I), che contiene le posizioni delle pecore.
Il ciclo I (100-150) disegna il campo di gioco e le pecore.
Le linee (170÷180) si occupano del testo e degli ingressi.
La linea 190 controlla la validità del colpo.
Il cambio giocatore è realizzato alla linea 220.

PER GIOCARE:

Si comincia l'esecuzione del programma con RUN. Si ottiene allora il messaggio GIOCATORE 1, che indica il numero del giocatore, seguito dalla domanda INIZIO,FINE?
Si risponde indicando il numero della pecora da spostare seguito da RETURN, poi il numero della casella di arrivo, seguito da RETURN.
Lo spostamento è allora realizzato; poi il secondo giocatore procede nello stesso modo. Il vincitore è in teoria quello che riesce a posizionare le sue tre pecore allineate sulle posizioni inizialmente occupate dall'avversario.

ESTENSIONI POSSIBILI:

Si comincerà con l'aggiungere un controllo del giocatore vincente. Si può anche migliorare la presentazione disegnando delle vere pecore, rendere il gioco più attraente numerandole, e imponendo un certo ordine all'arrivo. Finalmente, potremo creare una strategia per il calcolatore, fissando le regole.

```
00010 PRINT"La CAVALLINA 028@&"
00020 INIT 7:PAGE
00030 DIM A(8)
00040 A(1)=0:A(8)=0
00050 FOR I=1 TO 3
00060 A(I+1)=1
00070 A(I+4)=-1
00080 NEXT I
00090 H=1
00100 FOR I=1 TO 8
00110 CURSORX 4*I:CURSORY 8:PRINTI
00120 CURSORX4*I+1:CURSORY6:PRINT" "
00130 IFA(I)=1THENCURSORY 6:GOSUB 240
00140 IFA(I)=-1THENCURSORY 6:GOSUB 250
00150 NEXT I
00160 J=1-(H=-1)
00170 CURSORX 1:CURSORY 13
00175 PRINT"GIOCATORE ";J
00180 PRINT:INPUT "INIZIO , FINE :";D,F
00190 IFA(D)<>HORA(F)<>0THEN170
00200 A(D)=0
00210 A(F)=H
00220 H=-H
00230 GOTO 100
00240 CURSORX4*I+1:PRINT"M":RETURN
00250 CURSORX4*I+1:PRINT"W":RETURN
```

```
      M   M   M   W   W   W
  1   2   3   4   5   6   7   8


GIOCATORE  1

INIZIO , FINE :? 2,8
```

Lettere

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: media.

IL GIOCO:

La regola è fortemente ispirata da un gioco di lettere molto noto in FRANCIA, praticato notoriamente in una trasmissione televisiva (le cifre e le lettere).
Il Videopac sceglie a caso un certo numero di simboli (vocali e consonanti) nell'alfabeto latino, e ognuno dei giocatori dovrà indicare la lunghezza della parola che è riuscito ad ottenere (l'apparecchio non verificherà se la parola è corretta!).
Il punteggio totale di ogni giocatore è visualizzato in seguito.

IL PROGRAMMA:

Il ciclo I (90÷120) sceglie nell'alfabeto tra 7 e 12 lettere, in funzione della variabile aleatoria L, presa alla linea 60.
Per evitare di avere troppe consonanti, si utilizza in linea 110 una piccola astuzia, che consiste nel rifare a caso una presa nella tabella A$, contenente delle vocali (se ne può avere più o meno utilizzando delle altre RND).
Il ciclo (90÷120) chiede ad ognuno dei due giocatori la lunghezza della loro parola.
Le linee 190 e 200 paragonano i risultati, e totalizzano il punteggio di ogni giocatore, visualizzato alle linee 210 e 215.

PER GIOCARE:

Si lancia l'esecuzione del gioco con RUN. Le lettere appaiono allora, poi la domanda GIOCATORE 1 LUNGHEZZA? alla quale il primo giocatore risponde dando la lunghezza della parola che ha ottenuto (la validità della parola dovrà essere tuttavia controllata dall'altro giocatore); poi il giocatore 2 farà altrettanto ed il punteggio di ogni giocatore sarà allora visualizzato.
Per continuare la partita, basterà battere RET.
Si dovrà battere RESTORE per finire.

ESTENSIONI POSSIBILI:

Si potrebbe naturalmente pensare di introdurre una temporizzazione prima di chiedere le risposte ai giocatori. Si potrebbe anche introdurre non la lunghezza della parola, ma la parola stessa; l'avversario dovrà allora rispondere alla domanda CORRETTO O NO? Si potreb-

be anche estendere il gioco a più di due giocatori. E perché non cercare di definire un gioco contro il Videopac, nel quale sarà stata inserita prima una lista abbastanza lunga di parole, di cui visualizzerebbe le lettere nel disordine, con eventualmente qualche lettera di complemento scelta a caso?

```
00010 PRINT" - L E T T E R E - 029@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 S=O:J=0
00050 INIT 7:PAGE
00060 L=INT(RND(1)*6+7)
00070 A$="AEIOUEA"
00080 CURSORY 6:PRINT"LETTERE :"
00090 FOR I=1TO L:CURSORX8+2*I:CURSORY6
00100 PRINT CHR$(INT(RND(1)*26+65))
00110 IF RND(1)<.4 THEN GOSUB 240
00120 NEXT I
00130 I=1
00140 CURSORY 10:PRINT"GIOCATORE :";I;
00145 INPUT " -Lunghezza :";L
00150 CURSORX25:CURSORY10:PRINT"       "
00160 IFI=0 THEN P=L
00170 I=I+1
00180 IFI<3THEN140
00190 S=S+(PANDP>L)
00200 J=J+(LANDL>P)
00210 CURSORY13:PRINT"PUNTI - Giocato";
00215 PRINT"re 1:";S;" -Giocatore 2:";J
00220 PRINT:INPUT "Con RET continui";R$
00230 GOTO 50
00240 CURSORX 8+2*I:CURSORY 6:
00250 PRINTMID$(A$,INT(RND(1)*7+1),1)
00260 RETURN
```

```
LETTERE :E E A H B M A A

GIOCATORE : 2 -Lunghezza

PUNTI - Giocatore 1: 0 -Giocatore 2: 10

Con RET continui?
```

Macina parole

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: gioco di fortuna (sulle parole).
Difficoltà del gioco: nessuna.

IL GIOCO:

Questo cortissimo programma permette, al Videopac, di fare la dimostrazione delle sue capacità di manipolazione di stringhe di caratteri: si parte semplicemente da un prefisso e da un suffisso, di origine greca, che assembleremo a caso; cadremo naturalmente ogni tanto su delle parole assurde o anche leggermente equivoche. Una nuova parola è confezionata ad ogni tentativo, ma è possibile ottenere varie volte la stessa parola.

IL PROGRAMMA:

Due variabili A$ e B$ sono dimensionate a 15 nelle linee 50 e 60: una conterrà i prefissi, l'altra i suffissi (se si desidera aggiungere delle possibilità, basterà modificare questa limitazione a 15).
Il ciclo I (70÷90) legge le linee DATA (160÷220) riguardanti i prefissi ed i suffissi.
La linea 100 effettua la scelta di un prefisso a caso.
La linea 110 effettua la scelta di un suffisso a caso.
La parola ottenuta alla linea 120 è visualizzata dalla linea 130, poi si ricomincia (linea 150).

PER GIOCARE:

Lanciare l'esecuzione del programma con RUN. Una prima parola sarà immediatamente visualizzata, poi una nuova parola apparirà allora al seguito della precedente. Battere RESTORE per uscire dal programma.

ESTENSIONI POSSIBILI:

Si potrà facilmente aumentare la lista delle parole disponibili. Potremo anche utilizzare un processo analogo per fare creare delle frasi al calcolatore, fargli inventare delle poesie, ecc. Potremo anche trasformare il programma iniziale in un gioco di società che consisterebbe, per esempio, nel prendere delle scommesse sulla prossima parola che apparirà. E affidare, eventualmente, la gestione delle scommesse al Videopac...

```
00010 PRINT"MACINAPAROLE 030@&"
00020 INPUT "Dammi un Num.INTERO :";F.
00030 FOR I=1 TO R:Y=RND(1):NEXT I
00040 INIT 7
00050 DIM A$(15)
00060 DIM B$(15)
00070 FOR I=1 TO 15
00080 READA$(I),B$(I)
00090 NEXT I
00100 I=INT(RND(1)*15+1)
00110 J=INT(RND(1)*15+1)
00120 C$=A$(I)+B$(J)
00130 PRINTC$
00140 FOR S=1 TO1200:NEXT S
00150 GOTO 100
00160 DATA ANDRO,CRATE,COSMO,FILO,ORTO
00170 DATA FOBO,ELIO,TROPIO,SESSO,GEN0
00180 DATA NANO,PALEO,NAUTA,MACRO,FAGO
00190 DATA IDRO,MEGA,STENO,PATO,CARDIO
00200 DATA OLEO,DERMO,MITO,SCOPIO,OMEO
00210 DATA AUT0,FONO,AUDIO,VIDEO,FISIO
00220 DATA FUGO,MINI,LOGO,PSICO,STEREO
30760 ,40
```

```
COSMODERMO
OMEOPALEO
OMEOIDRO
ELIOCRATE
SESSOSTENO
ANDROMACRO
FAGOCARDIO
ORTOTROPIO
FONOCRATE
NAUTAAUT0
VIDEOGEN0
FAGOSTENO
OMEOTROPIO
VIDEOCARDIO
COSMOCRATE
ANDROAUT0
ELIOMACRO
PATOGEN0
MEGAAUT0
```

Spia

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: gioco di riflessione.
Difficoltà del gioco: facile.

IL GIOCO:

Il programma può servire simultaneamente da codifica e decodifica di messaggi segreti, seguendo un metodo molto facile da mettere in opera su questo tipo di calcolatore.
Un piccolo generatore aleatorio situato alla fine del programma è utilizzato per modificare in maniera arbitraria ogni lettera del messaggio; si dovrà fare attenzione ad utilizzare sempre lo stesso numero (nucleo) per cominciare la codifica o la decodifica di due messaggi identici.

IL PROGRAMMA:

Il nucleo iniziale (che sarà un intero qualunque) è inserito alla linea 20; il messaggio è inserito alla linea 40.
La linea 50 permette di scegliere tra le opzioni di codifica e decodifica.
Il ciclo I (80÷150) è utilizzato per effettuare sia la codifica che la decodifica.
La linea 100 chiama il generatore aleatorio.
La linea 110 realizza la trasposizione delle lettere.
Le linee 120 e 130 verificano che si resti nei limiti autorizzati.
I risultati saranno stampati dalla linea 160.
Le linee da 200 a 230 contengono il generatore.

PER GIOCARE:

Lanciare il programma con RUN. Inserire in seguito un intero N che servirà da nucleo per il generatore (bisognerà ricordarselo per la decodifica). Poi battere il messaggio, seguito da RETURN. Scegliere l'opzione C per CODIFICARE, e l'opzione D per DECODIFICARE, e battere RETURN.
Il messaggio codificato o decodificato, secondo il caso, sarà allora visualizzato.

ESTENSIONI POSSIBILI:

Si potrà estendere la codifica ad altri segni che le lettere dell'alfabeto, utilizzare eventualmente vari generatori, introdurre nel messaggio una codifica del o dei numeri che sono serviti da generatori, ecc.

```
00010 PRINT"La SPIA 031@&":PRINT
00020 INPUT"Dammi il tuo CODICE :";N
00030 A=1/N:PRINT
00040 INPUT"Il tuo MESSAGGIO:";A$:PRINT
00050 PRINT"(C)ODIFICA/(D)ECODIFICA :";
00060 INPUT R$
00070 B$=""
00080 FOR I=1 TO LEN(A$)
00090 T=ASC(MID$(A$,I,1))
00100 GOSUB 200
00110 T=T-A*((R$="C")-(R$="D"))
00120 IFT>90THENT=T-26
00130 IFT<65THENT=T+26
00140 B$=B$+CHR$(T)
00150 NEXT I
00160 PRINT:PRINT"RISULTATO :";B$
00170 SOUND5:SOUND5:SOUND5:SOUND4:PRINT
00180 SOUND4:SOUND2:SOUND0:PRINT:GOTO10
00200 A=A(A+3.14159) 8
00210 A=A-INT(A)
00220 A=INT(26*A+1)
00230 RETURN
30010 PRINT"La SPIA 031@&":PRINT
```

```
Ok
run
La SPIA  031@&

Dammi il tuo CODICE :? 123

Il tuo MESSAGGIO:? GRANDE SLAM

(C)ODIFICA/(D)ECODIFICA :? C

RISULTATO :HSBOEF:TMBN
```

Agganciate i vagoni

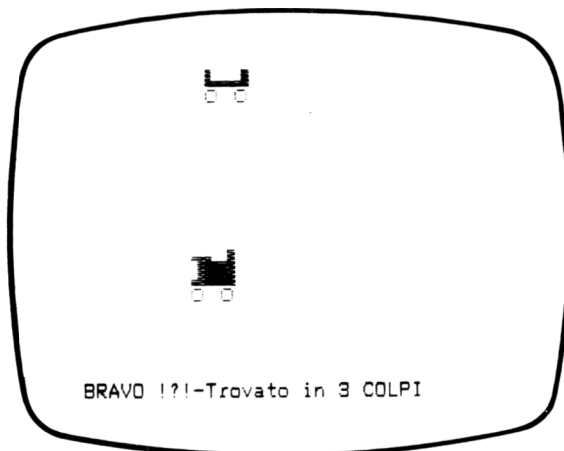
RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: solitario.
Difficoltà del gioco: facile.

IL GIOCO: Bisogna cercare di agganciare, il più rapidamente possibile, la locomotiva con il suo vagone, che appaiono tutti e due all'inizio della partita, in due posti più o meno lontani dello schermo. Dovremo per questo spostare la locomotiva indicando il seguito dei movimenti da effettuare (in maniera codificata: A per una casella verso l'alto; B per una casella verso il basso; D per una casella verso destra; e S per una casella verso sinistra). Si batterà, per esempio: BBSDAAA.

IL PROGRAMMA: Le linee 40 e 50 disegnano la locomotiva, la linea 60 il vagone; le ruote sono alla linea 70.
La posizione iniziale della locomotiva è definita dalle linee 90 e 100, quella del vagone dalle linee 110 e 120.
La linea 130 evita che siano inizialmente troppo vicini.
Le linee 150÷170 realizzano la visualizzazione del vagone e le linee 180 e 190 quella della locomotiva.
Il test di fine è in linea 200.
I movimenti da effettuare sono inseriti, in una variabile alfanumerica R$, alla linea 215, e sono decodificati in un ciclo I (240÷270), per essere eseguiti alla linea 280.

PER GIOCARE: RUN permette di visualizzare le posizioni di partenza della locomotiva e del vagone. Si batte allora la serie degli spostamenti da effettuare, sotto la forma DDDAAABASS, seguito da RETURN. La locomotiva apparirà nella sua nuova posizione, e si ricomincia, fino a che non sia attaccata al vagone. Si otterrà allora l'apparizione del punteggio.

ESTENSIONI POSSIBILI: Sarà interessante prevedere la perdita della partita se, per incidente, nel corso dello spostamento, la locomotiva entra in collisione col vagone.
Potremo anche prevedere vari vagoni, e un ordine di aggancio, giocare in molti con varie locomotive, disegnare le ferrovie, introdurre degli ostacoli da aggirare obbligatoriamente da una certa parte, ecc.

```
00010 PRINT"Agganciate i VAGONI 032@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 D$=CHR$(108)+CHR$(116)+CHR$(122)
00050 A$=CHR$(122)+CHR$(127)+CHR$(127)
00060 B$=CHR$(117)+CHR$(112)+CHR$(122)
00070 C$=CHR$(79)+CHR$(96)+CHR$(79)
00080 N=0:PAGE
00090 LY=INT(RND(1)*33)
00100 LX=INT(RND(1)*20+1)
00110 WY=INT(RND(1)*33)
00120 WX=INT(RND(1)*20+1)
00130 IFABS(LX-WX)<4GOTO 90
00140 INIT 7:GR0,7,0
00150 CURSORX WY:CURSORY WX:PRINTB$
00160 CURSORX LY:CURSORY LX-1:PRINTD$
00170 CURSORX LY:CURSORY LX:PRINTA$
00175 TX0,0,0
00180 CURSORX WY:CURSORY WX+1:PRINTC$
00190 CURSORX LY:CURSORY LX+1:PRINTC$
00200 IFLX=WXANDWY=LY-3GOTO 300
00205 CURSORX 1:CURSORY 19:PRINT"PER";
00210 PRINT"CORSO da Seguire"
00215 INPUT "(dssabb....) :";R$
00220 X=0:Y=0
00230 N=N+1
00240 FOR I=1 TO LEN(R$)
00250 X=X-(MID$(R$,I,1)="b"ANDLX<17)
00255 X=X+(MID$(R$,I,1)="a"ANDLX>1)
00260 Y=Y-(MID$(R$,I,1)="d"ANDLY<37)
00265 Y=Y+(MID$(R$,I,1)="s"ANDLY>1)
00270 NEXT I
00280 LX=LX+X:LY=LY+Y
00290 GOTO 140
00300 CURSORX1:CURSORY20:PRINT"BRAVO ";
00310 PRINT"!?!-Trovato in";N;" COLPI"
00320 SOUND5:SOUND2:SOUND5:SOUND4:END
```

Lancio di un dado truccato

RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: riflessione.
Difficoltà del gioco: media.

IL GIOCO: Riprende un'idea di un programma precedente (DADO TRUCCATO) in un'applicazione statistica: si fabbrica un dado volontariamente truccato, che si lancerà un certo numero di volte, e si crea una tabella di risultati.

IL PROGRAMMA: Le probabilità desiderate per ogni tiro sono introdotte da un ciclo I (70÷140), con tutte le verificazioni di validità necessarie. Il numero di tiri richiesti è introdotto alla linea 160.
Il ciclo J (170÷250) contabilizza i lanci.
Il ciclo I (190÷210) assicura la conformità di ogni lancio con le probabilità introdotte all'inizio del gioco.
I risultati sono contabilizzati alla linea 220, e visualizzati grazie alle linee 230 e 240.

PER GIOCARE: Cominciare l'esecuzione battendo RUN, poi, alla domanda PROBABILITÀ DEL DADO 1, introdurre un numero tra 0 e 1. Fare lo stesso per i dadi 2, 3, ecc., senza dimenticare che il totale delle probabilità introdotte dovrà essere vicinissimo a 1. In seguito, all'apparizione della domanda NUMERO DI LANCI si introdurrà un intero sufficientemente grande, seguito da RETURN. La visualizzazione dei risultati richiederà tanto più tempo ad apparire quanto più è grande l'intero inserito. I risultati sono visualizzati sotto i numeri dei lanci, che appaiono in caratteri reverse: così 32.3 apparendo sotto l'uno reverse significa che il risultato 1, al quale avevamo assegnato una probabilità di 0.3, è stato realizzato 32 volte.

ESTENSIONI POSSIBILI: Possiamo immaginare di migliorare la raffigurazione dei risultati utilizzando per esempio un istogramma, oppure un qualsiasi altro metodo graficamente più soddisfacente.

```
00010 PRINT"Tiro/DADO Truccato 033@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 DIM E(6)
00050 INIT 7
00060 P=0
00070 FOR I=1 TO 6
00080 CURSORX 1:CURSORY 3:PRINT"          ";
00085 PRINT"                    "
00090 CURSORX1:CURSORY3:PRINT"PROBABI";
00095 PRINT"LITA'del DADO";I;:INPUTE(I)
00100 P=P+E(I)
00110 IFE(I)<0ORE(1)>1ORP>1.1THEN300
00120 CURSORY 6:PRINTSPC(6*I-5);
00125 PRINT"DADO ";I
00130 CURSORY 8:PRINTSPC(6*I-5);E(I)
00140 NEXT I
00150 IFP<.9THENRUN:PRINT
00160 INPUT "Numero di TIRI :";P
00170 FOR J=1 TO 6
00180 N=INT(RND(1)*6+1)
00190 FOR I=1 TO 6
00199 IFN=1GOTO 180
00200 IFRND(1)>(E(I)-INT(E(I))GOTO 210
00205 IFN=1GOTO 180
00210 NEXT I
00220 E(N)=E(N)+1
00230 CURSORX 6*N-4:CURSORY 15:PRINTN
00240 CURSORX 6*N-5:CURSORY 17PRINTE(N)
00250 NEXT J
00260 SOUND4:SOUND2:SOUND5:SOUND2:END
00300 CURSORX 1:CURSORY 4
00310 PRINT"IMPOSSIBILE :(0<=P<=1)"
00320 P=P-E(I)
00330 FOR J=1 TO 1700:NEXT J
00340 CURSORX 1:CURSORY 4:PRINT"          ";
00350 PRINT"        ":GOTO 90
```

```
PROBABILITA'del DADO 1? .1

DADO  1

 .1
```

Giro di carte

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco di prestidigitazione.
Difficoltà del gioco: facile.

IL GIOCO:

È un gioco abbastanza facile ma, qui, le carte sono manipolate sotto i vostri occhi dal Videopac che ha il ruolo di prestigiatore; 21 carte, prese a caso in un mazzo di 52, sono visualizzate, e il giocatore dovrà scegliere mentalmente una delle carte proposte. Poi, dovrà indicare al Videopac in quale colonna (1ma, 2nda, o 3za partendo da sinistra) si trova la carta scelta; le carte sono allora ridistribuite ed il giocatore indica di nuovo la colonna corrispondente alla sua carta; questo è fatto di nuovo ancora una volta; alla quarta visualizzazione, il Videopac è in grado di indovinare la carta scelta.

IL PROGRAMMA:

Il ciclo I (60÷120) permette di scegliere, a caso, 21 carte da 52, ed il ciclo chiuso J (80÷100) permette di assicurarsi che non si realizzi due volte la stessa scelta.
La linea 140 contiene il colore delle carte, così come il nome delle figure (FA, DA, RE, AS), in una variabile alfanumerica B$.
Il ciclo I (150÷240) realizza la visualizzazione del gioco.
La linea 160 decodifica il colore della carta, la linea 170 sceglie la posizione di visualizzazione, la linea 180 decodifica l'altezza.
La visualizzazione vera e propria è realizzata dalla linea 190 alla linea 235.
La linea 260 permette, al giocatore, di inserire il numero della colonna dove si trova la carta.
Il ciclo J (300÷320) realizza la mescola, ed i cicli chiusi J (350÷390) e I (360÷380) fabbricano la nuova variabile A$ da utilizzare per la prossima visualizzazione.

PER GIOCARE:

Si lancia il programma con RUN. Appaiono le prime 21 carte, divise in tre colonne. Si indica allora il numero (1, 2, 3) della colonna dove si trova la carta scelta, seguito da RETURN.
Si ottiene una seconda visualizzazione e si procede nella stessa maniera; poi una terza e si fa la stessa cosa. Alla quarta mano, la carta scelta è automaticamente visualizzata.
Si uscirà dal programma con RESTORE al posto del numero di colonna.

ESTENSIONI POSSIBILI: Si può immaginare una miglioria della visualizzazione scrivendo il nome delle carte per intero, o anche disegnando delle vere carte, a colori.
Si potrà anche cercare di programmare degli altri giochi, usando lo stesso principio base, ma con più carte, ecc.

```
00010 PRINT"Giro di CARTE 034@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 A$="":CLEAR500
00060 FOR I=1 TO 21
00070 B$=CHR$(INT(RND(1)*52+38))
00080 FOR J=1 TO LEN(A$)
00085 C$=MID$(A$,J,1)
00090 IFB$=C$THENJ=LEN(A$):NEXTJ:GOTO70
00100 NEXT J
00110 A$=A$+B$
00120 NEXT I
00130 H=3:P=1
00140 B$="FiQuCuPiFADAREAS"
00150 FOR I=1 TO LEN(A$)*P-11*(P=0)
00160 S=INT((ASC(MID$(A$,I,1))-38)/13)
00170 J=7*INT((I-1)/7)
00180 T=ASC(MID$(A$,I,1))-37-13*S
00190 IFP<>0THENGOTO 205
00195 CURSORX1:CURSORY5:PRINT"La tua ";
00200 PRINT"CARTA e':                 "
00205 CURSORX J*P+INT(J/2)-16*(P=0)+1
00210 CURSORY (I-J)*P+5
00215 IFT<=9THENPRINTSTR$(T+1);:GOTO225
00220 PRINTMID$(B$,ABS(2*T-11),2);
00225 Z=(J+3)*P+1+INT(J/2)-21*(P=0)
00230 CURSORX Z:CURSORY (I-J)*P+5
00235 PRINTMID$(B$,2*S+1,2)
00240 NEXT I
00250 IFP=0THENSOUND4:SOUND2:END
00255 CURSORX 1:CURSORY 16
00260 INPUT "Che COLONNA :";N
00270 INIT 7
00280 H=H-1
00290 R$="20120":B$=""
00300 FOR J=0 TO 2
00305 K=7*VAL(MID$(R$,N+J,1))
00310 B$=B$+MID$(A$,K+1,7)
00320 NEXT J
00330 IFH=0THENA$=B$:P=0:GOTO 140
00340 A$=""
00350 FOR J=0 TO 2
00360 FOR I=1 TO 7
00370 A$=A$+MID$(B$,3*I-2+J,1)
00380 NEXT I
00390 NEXT J
00400 GOTO 140
```

Il gioco di Marienbad

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: logica (2 giocatori).
Difficoltà del gioco: media.

IL GIOCO:

Un gioco che è stato reso molto celebre grazie ad un film, tuttavia presentato qui in una versione più conforme alle sue origini, ossia il gioco di NIM.
Infatti, è possibile, all'inizio della partita, scegliere il numero di mucchi che vogliamo fare ed il numero di fiammiferi che vogliamo mettere in ogni mucchio. Ogni giocatore gioca al suo turno, e quello che prende l'ultimo fiammifero ha perso.

IL PROGRAMMA:

La linea 60 dimensiona una variabile A (I) che conterrà il numero di fiammiferi di ogni mucchio.
Il ciclo I (70÷100) permette di inserire questo numero.
Il ciclo I (130÷160) stampa il mucchio.
Le linee 170 e 175 garantiscono la visualizzazione del numero del giocatore.
La linea 190, così come la 200, controlla se la mossa è autorizzata.
La linea 270 controlla la fine della partita.

PER GIOCARE:

Si lancia il programma con RUN, poi si inserisce il numero di mucchi (tra 1 e 8), seguito da RETURN. Si inserirà in seguito il numero di fiammiferi in ogni mucchio (fare ad ogni ingresso N seguito da RETURN). Si otterrà allora la visualizzazione dei differenti mucchi, col numero del giocatore interessato (1 o 2), così come il messaggio IN QUALE MUCCHIO?; si risponde battendo il numero del mucchio nel quale si desidera pescare seguito da RETURN. Si ottiene allora la domanda QUANTITÀ DA TOGLIERE?, e si risponde indicando il numero di fiammiferi che si desidera levare dal mucchio. Poi sarà il turno del secondo giocatore, e si giocherà alternativamente fino a che non ci siano più fiammiferi disponibili.
In fine partita, si otterrà la visualizzazione del numero del giocatore vincente.

ESTENSIONI POSSIBILI:

Si può designare ogni giocatore col suo nome, fare un grafismo più elaborato, scrivere un programma che permetta al Videopac di fare da partner, ecc.

```
00010 PRINT"Gioco di MARIENBAD 0350&"
00020 SOUND4:SOUND0:SOUND2:SOUND5
00030 INPUT "Numero di MUCCHI (<=7):";N
00040 IFN>=8THENRUN
00050 S=0
00060 DIM A(N)
00070 FOR I=1 TO N
00080 PRINT:PRINT"MUCCHIO";I;"  -  ";
00085 INPUT "Num.di FIAMMIFERI :";A(I)
00090 S=S+A(I)
00100 NEXT I
00110 H=1
00120 INIT 7
00130 FOR I=1 TO N
00140 CURSORX 5*I+3:CURSORY 4:PRINTA(I)
00150 CURSORX1:CURSORY6:PRINT"MUCCHIO"
00155 CURSORX 5*I+3:CURSORY 6:PRINTI
00160 NEXT I
00170 CURSORX1:CURSORY10
00175 PRINT"Giocatore ";H
00180 PRINT:INPUT "In che MUCCHIO :";T
00190 IFT>NGOTO 120
00200 IFA(T)=0GOTO 120
00210 CURSORX 1:CURSORY 14
00215 INPUT "QUANTITA' da Togliere :";O
00220 IFO>A(T)ORO=0GOTO 200
00230 A(T)=A(T)-O
00240 S=S-O
00250 H=H+1
00260 IFH=3THENH=1
00270 IFS<>0GOTO 120
00275 PRINT:PRINT"Il Giocatore ";
00280 PRINTH;" ha VINTO !?!"
00290 SOUND0:SOUND2:SOUND4:SOUND5:END
```

```
Ok
run30
Numero di MUCCHI (<=7):? 7

MUCCHIO 1   -   Num.di FIAMMIFERI :? 7

MUCCHIO 2   -   Num.di FIAMMIFERI :? 29

MUCCHIO 3   -   Num.di FIAMMIFERI :? 23

MUCCHIO 4   -   Num.di FIAMMIFERI :? 13

MUCCHIO 5   -   Num.di FIAMMIFERI :? 19

MUCCHIO 6   -   Num.di FIAMMIFERI :? 17

MUCCHIO 7   -   Num.di FIAMMIFERI :? 11
```

```
          7    29   15   6    19   7    7

MUCCHIO 1    2    3    4    5    6    7


Giocatore  2

In che MUCCHIO :? 2

QUANTITA' da Togliere :? 12
```

L'indesiderabile

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: caso e riflessione (1 giocatore).
Difficoltà del gioco: media.

IL GIOCO:

Siete il proprietario di un immobile di 16 vani vuoti e non riuscite a mettere la mano su un vostro inquilino poco desiderabile. Visitate i vani a caso, ma, se visitate la camera accanto, il vostro abusivo, messo in guardia dal rumore, sgombra rapidamente. Egli può addirittura risistemarsi in un vano già visitato e accertato come vuoto. Bisognerà riuscire a sloggiarlo in meno di 9 colpi per vincere.

IL PROGRAMMA:

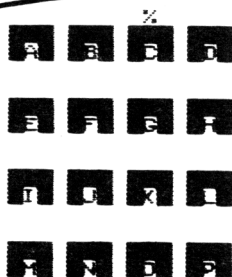
Una tabella viene dimensionata a 16 valori, alla linea 60, per ricevere il numero delle caselle già visitate.
Il ciclo I (80-110) traccia il disegno del gioco.
La posizione iniziale dell'abusivo è estratta a sorte alla linea 115.
Il numero della casella da visitare è dato alla linea 130.
Le linee da 145 a 170 controllano le varie risposte possibili.
Il ciclo I (180-200) disegna le caselle visitate.
Le linee 240 e 245 disegnano l'inquilino che viene cancellato alla linea 275.

PER GIOCARE:

Si lancia il programma con RUN.
Alla domanda "Casella VISITATA", si risponde battendo una lettera da A a P. Se si è trovato l'abusivo, esso viene visualizzato; altrimenti si possono verificare due casi: o si è abbastanza lontani e l'abusivo non si muove, oppure è nella casella, appare per un breve istante, e si va a rinascondere, casualmente, in una casella vicina.
Le caselle già visitate si colorano e non potete ritornarvi dentro. Se al completamento dei 9 colpi, non l'avete trovato, la posizione dell'abusivo viene evidenziata e avete perso.

ESTENSIONI POSSIBILI:

Naturalmente, si potrà giocare su più di 16 caselle, utilizzare dei caratteri grafici per rappresentare il proprietario e l'inquilino, usare un algoritmo più sofisticato per lo spostamento dell'abusivo, vietargli eventualmente di ritornare nelle caselle già occupate in precedenza, ecc...

```
00010 PRINT"L' INDESIDERABILE 03606"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 A$=CHR$(127)
00050 B$=A$+A$+A$
00060 DIM B(16)
00070 INIT 7
00080 FOR I=1 TO 16
00090 CURSORX 4+4*(I-1-4*INT((I-1)/4))
00095 CURSORY 2+4*INT((I-1)/4):PRINTB$
00100 CURSORX 4+4*(I-1-4*INT((I-1)/4))
00105 CURSORY 3+4*INT((I-1)/4):PRINTA$;
00110 PRINTCHR$(64+I);A$:NEXT I
00115 X=INT(RND(1)*16+1):S=1:K=0
00120 IFS=1THEN:TX0,4,1:PRINT"Fa' LOCK"
00125 CURSORX10:CURSORY17:TX0,0,0
00130 INPUT "CASELLA Visitata :";C$
00135 C=ASC(C$)-64
00140 CURSORX 1:CURSORY 19
00145 IFB(C)=1THENPRINT"Gia' ";:GOTO300
00150 IFC=XTHENPRINT"L'hai ";:GOTO310
00160 IFX-5=CORX-4=CORX-3=CGOTO 240
00165 IFX-1=CORX+1=CORX+3=CGOTO 240
00170 IFX+4=CORX+5=CGOTO 240
00175 PRINT"NON e'la'":SOUND 5:SOUND 5:
00180 S=S+1:B(C)=1:FOR I=1 TO 16
00190 CURSORX 4+4*(I-1-4*INT((I-1)/4))
00195 CURSORY 2+4*INT((I-1)/4):PRINTB$
00200 NEXT I
00210 IFS<>9GOTO 230
00215 CURSORX1:CURSORY19:PRINT"Hai ";
00220 PRINT"PERSO era in ":CHR$(X+64)
00225 K=1:GOTO 240
00230 CURSORX1:CURSORY19:PRINT"       ";
00235 PRINT"                    ":GOTO120
00240 CURSORX 5+4*(X-1-4*INT((X-1)/4))
00245 CURSORY 1+4*INT((X-1)/4):PRINT"%"
00250 FOR I=1 TO 300:NEXT I
00260 IFK=1THENSOUND4:SOUND4:SOUND5:END
00270 CURSORX 5+4*(X-1-4*INT((X-1)/4))
00275 CURSORY 1+4*INT((X-1)/4):PRINT" "
00280 Z=INT(10*RND(1)+1)-5
00285 IFABS(Z)=2GOTO 280
00290 X=X+Z:IFX<=16ORX>=1GOTO 180
00295 X=X-Z:GOTO 280
00300 PRINT"VISITATA":SOUND2:GOTO 125
00310 PRINT"TROVATO in ";S;" COLPI.";
00320 PRINT"BRAVO !!?!!";SOUND5:SOUND4
00330 SOUND4:SOUND5:SOUND2:K=1:GOTO 240
```

Componi

RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: composizione musicale.
Difficoltà del gioco: nessuna, a priori.

IL GIOCO: Si tratta, infatti, di un programma capace di comporre, in modo molto elementare, della musica, e questo in una maniera completamente aleatoria. Il calcolatore, in un primo tempo, disegnerà uno spartito, poi stampa una successione di note (che vanno dal Do grave al Si acuto) e dà il loro nome. Una nuova serie di note apparirà in seguito, e così via.

IL PROGRAMMA: Il ciclo I di linea 65 disegna lo spartito.
La variabile N$, alla linea 80, contiene i nomi delle note.
Il secondo ciclo I (90-170) permette di fare la scelta di una nota in maniera aleatoria, alla linea 100; il suo nome è calcolato alla linea 110 e la nota è disegnata alle linee (120÷140).
La linea 190 permette di ricominciare battendo RET.

PER GIOCARE: Lanciare l'esecuzione del programma battendo RUN. Lo spartito sarà allora molto rapidamente visualizzato, poi le note sono rappresentate, col loro nome in basso sullo schermo. Una pressione sul tasto RET provocherà l'apparizione di una nuova serie di note, ed una pressione sul tasto RESTORE provoca la fine della partita.

ESTENSIONI POSSIBILI: Potremo affinare le regole di composizione, visualizzare più note, introdurre dei diesis e dei bemolle, migliorare il grafismo, fare suonare la melodia visualizzata.

```
00010 PRINT"COMPOSIZIONE 037@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F=Y=RND(1):NEXT I
00040 INIT 7
00050 FOR I=1 TO5
00060 CURSORX 1:CURSORY 3+2*I
00065 FORJ=1TO38:PRINTCHR$(96);:NEXTJ
00070 NEXT I
00080 N$="SI LA SOLFA MI RE DO"
00090 FOR I=1 TO 13
00100 X=INT(RND(1)*14+1)
00110 D$=MID$(N$,3*(X+7*(X>7))-2,3)
00120 CURSORX 3*I-2:CURSORY X-1
00125 PRINTCHR$(125):CURSORX 3*I-2
00130 CURSORY X:PRINTCHR$(125)
00135 CURSORX3*I-2:CURSORYX+1:PRINT"O"
00140 CURSORX 3*I-2:CURSORY 18:PRINTD$
00150 IFX<=7THENSOUND4:SOUND2:GOTO 170
00160 SOUND 2:SOUND 6:SOUND 4:SOUND 3
00170 NEXT I
00180 CURSORX 1:CURSORY 20
00190 INPUT "Con-RET-continui";R$
00200 GOTO 40
```

```
LA SI SI MI SOLDO SI FA FA MI FA LA LA

Con-RET-continui?
```

Giorno della settimana

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: utilitaria.
Difficoltà del gioco: facile.

IL GIOCO:

Il programma vi permette di ritrovare, senza alcun calcolo, il giorno della settimana corrispondente ad una certa data.
La data sarà inserita sotto forma di tre numeri, per esempio 19, 11, 1967 per il 19 novembre 1967, ed il risultato apparirà.

IL PROGRAMMA:

Le linee da 10 a 70 controllano gli inserimenti.
La linea 35 si occupa della fine della partita.
La linea 90 permette una correzione per le date precedenti il 1582 (messa in opera dell'attuale calendario).
Le linee (100÷120) permetteranno la visualizzazione in lettere del mese nella risposta (su 3 lettere).
La formula utilizzata appare alle linee 140 e 150.
Le linee (160÷180) permetteranno la visualizzazione del giorno nella risposta.
La linea 210 stampa i risultati.
Il sottoprogramma 250 interviene per tener conto dei mesi di gennaio e febbraio.

PER GIOCARE:

Lanciare il programma con RUN, poi inserire il giorno seguito da RETURN, il mese (tra 1 e 12) seguito da RETURN, e l'anno (nella forma 1845, per esempio) seguito da RETURN. Il risultato è rapidamente visualizzato ed il programma ritorna sulla domanda GIORNO. Per continuare, basta ribattere un giorno, un mese, un anno; per terminare, bisogna rispondere 0 alla domanda GIORNO.

ESTENSIONI POSSIBILI:

Si potrà inserire il mese in lettere e fare visualizzare il nome del giorno per intero. Si può immaginare di fare apparire le date delle feste particolari (Natale, Pasqua, Pentecoste...).
Si può scrivere un programma che visualizzerebbe i giorni festivi per un dato anno, ecc.

```
00010 PRINT"GIORNO di Settimana 038@&"
00020 PRINT:CLEAR200
00030 PRINT:INPUT "GIORNO...:";G
00040 PRINT:INPUT "MESE.....:";M
00050 PRINT:INPUT "ANNO.....:";A
00060 IFG=0ORM=0THENPRINT"Errato!?":END
00070 B=A
00080 S=INT(A/100)-INT(A/400)
00090 IFA<=1582THENS=2
00100 M$="GEN.FEB.MAR.APR.MAG.GIU.LUG."
00110 M$=M$+"AGO.SET.OTT.NOV.DEC."
00120 M$=MID$(M$,4*M-3,4)
00130 IFM<3THENGOSUB 250
00140 Z=G-1+INT(13*(M+1)/5)+INT(5*A/4)
00150 S=Z-S-7*INT((Z-S)/7)
00160 G$="DOMENICA LUNEDI   MARTEDI  "
00170 G$=G$+"MERCOLEDIGIOVEDI  VENERDI"
00180 G$=G$+"  SABATO   "
00190 G$=MID$(G$,9*S+1,9)
00200 CURSORX1:CURSORY15:PRINT"Il ";G;
00210 PRINT" ";M$;B;" cade di ";G$
00220 SOUND 4:SOUND 2:SOUND 4:SOUND 4
00230 PRINT:INPUT "Con-RET-continui";R$
00240 INIT 7:GOTO 30
00250 A=A-1:M=M+12:RETURN
```

```
GIORNO...:? 1

MESE.....:? 1

ANNO.....:? 1985




Il  1 GEN. 1985 cade di SABATO

Con-RET-continui?
```

Il gioco della vita

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco solitario (2 versioni).
Difficoltà del gioco: facile.

IL GIOCO:

La versione presentata qui rispetta le regole poste all'origine dal matematico J.H. CONWAY.
Una certa popolazione è definita all'inizio del gioco sia dal Videopac (versione automatica) sia da voi stessi (versione manuale). Questa popolazione evolverà sotto i vostri occhi, rispettando certe regole, molto elementari, di nascita, di vita e di morte. L'evoluzione proseguirà fino a che non si deciderà di fermare il programma.

IL PROGRAMMA:

Due tabelle (B e C) sono utilizzate alla linea 50, per registrare lo stato della popolazione.
I cicli F (230-310) e G (240-300) sono utilizzati all'inizio del gioco per definire la popolazione iniziale (linee 250 e 260).
Il calcolo della nuova popolazione è in seguito realizzato con l'aiuto di un ciclo quadruplo: 80-210, 90-200, 110-160, 120-250.
La linea 130 realizza la gestione del nuovo stato secondo le regole.
La visualizzazione di questo nuovo stato è ottenuta riutilizzando il doppio ciclo di visualizzazione F (230-310) e G (240-300).
La linea 320 permette di ricominciare.

PER GIOCARE:

Nella versione automatica, basta lanciare il programma con RUN e guardare.
Per fermarsi, bisognerà battere RESTORE.
Nella versione manuale, la configurazione di inizio sarà inserita, dopo il lancio del programma con RUN, battendo 0 se non si vuole mettere niente sulla linea e sulla colonna indicate, e 1 se si desidera metterci una cellula. Poi l'evoluzione si svolgerà in seguito automaticamente.

ESTENSIONI POSSIBILI:

Sarà facile, se si dispone di una memoria più importante, di aumentare le dimensioni delle tabelle B e C, ridefinendo la variabile A. Si potrà anche introdurre il colore, migliorare la precisione del disegno rimpiazzando i PRINT con una visualizzazione ad alta risoluzione.

```
00010 PRINT"Gioco della VITA-A 0390&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INPUT "DIMENSIONE :";A
00050 DIM B(A,A):DIM C(A,A)
00060 I=0
00070 GOTO 220
00080 FOR F=1 TO A
00090 FOR G=1 TO A
00100 S=0
00110 FOR I=-1 TO 1
00120 FOR J=-1 TO 1
00130 IFF+J=0ORG+I=0GOTO 150
00135 IFF+J=A+1ORG+I=A+1GOTO 150
00140 S=S+B(F+J,G+I)
00150 NEXT J
00160 NEXT I
00170 S=2*S-B(F,G)
00180 C(F,G)=0
00190 IFS>=5ANDS<=7THENC(F,G)=1
00200 NEXT G
00210 NEXT F
00220 INIT 7
00230 FOR F=1 TO A
00240 FOR G=1 TO A
00250 IFI<>0GOTO 270
00260 IFRND(1)<.5THENC(F,G)=1
00270 CURSORX 2*G:CURSORY 2*F
00280 IFC(F,G)=1THENPRINT"*"
00290 B(F,G)=C(F,G)
00300 NEXT G
00310 NEXT F
00320 GOTO 80
```

```
LINEA  5 COLONNA 5 (Batter 1 o 0) ? 0

.
        .

 * *   * *
 *
         *
 *       *
 *   * *
```

```
00010 PRINT"Gioco della VITA-M 0400&"
00040 INPUT "DIMENSIONE :";A
00050 DIM B(A,A):DIM C(A,A)
00060 I=0
00070 GOTO 220
00080 FOR F=1 TO A
00090 FOR G=1 TO A
00100 S=0
00110 FOR I=-1 TO 1
00120 FOR J=-1 TO 1
00130 IFF+J=0ORG+I=0GOTO 150
00135 IFF+J=A+1ORG+I=A+1GOTO 150
00140 S=S+B(F+J,G+I)
00150 NEXT J
00160 NEXT I
00170 S=2*S-B(F,G)
00180 C(F,G)=0
00190 IFS>=5ANDS<=7THENC(F,G)=1
00200 NEXT G
00210 NEXT F
00220 INIT 7
00230 FOR F=1 TO A
00240 FOR G=1 TO A
00250 IFI<>0GOTO 270
00255 CURSORX1:CURSORY18:PRINT"LINEA ";
00260 PRINTF;" COLONNA";G;" (Batter 1";
00265 PRINT" o 0) ";:INPUT C(F,G)
00270 CURSORX 2*G:CURSORY 2*F
00280 IFC(F,G)=1THENPRINT"*"
00290 B(F,G)=C(F,G)
00300 NEXT G
00310 NEXT F
00320 GOTO 80
```

```
* * * * *
  *   *
    * * *
*   *   *
* * *
```

LIVELLO 3

N.	Nome del programma	Tema e particolarità
41	Un dado	Lancio d'un dado, con grafica.
42	Più grande, più piccolo	Interessante gioco di deduzione per un giocatore.
43	Ordine	Un po' d'astuzia per metter ordine il più in fretta possibile.
44	L'impiccato	Classico gioco di lettere, per due giocatori.
45	Briscola-bellotta	Simulazione d'una mano di 32 carte.
46	Puzzle di calcolo	Buon allenamento per due giocatori.
47	Impiccato inverso	Tocca ora all'elaboratore indovinare la parola.
48	Nicomaco	Per perfezionarsi in calcolo mentale, uno o più giocatori.
49	Corsa di cavalli	Gioco d'azzardo, assai visivo, per uno o più giocatori.
50	Attraverso	Gioco di strategia, per due giocatori.
51	Mastermind	Un classico con i numeri, per un solo giocatore.
52	Idem	Gioco di calcolo mentale a più livelli, per un solo giocatore.
53	Orologio	Non buttate via il vecchio televisore, trasformatelo in un orologio.
54	Regine della scacchiera	Un vecchio problema per aiutarvi a riflettere.
55	Ripetizioni	Rintracciate una sequenza di numeri sempre più lunga.
56	Parole	Un'idea semplicissima ma non così facile da indovinare.
57	Permutazioni	Gioco di riflessione non sempre evidente.
58	Bridge	Per metter a punto i vostri problemi di bridge.
59	Hanoi	Un consiglio! Cominciate con il piccolo.
60	Farfalle	Per farvi prender sonno.
61	Matteo	Non sapete disegnare? Lasciate fare all'elaboratore.

Manipolazione di stringhe

Caratteri speciali e stato dei caratteri

TRATTAMENTO DI STRINGHE DI CARATTERI: MID$, LEFT$, RIGHT$

Abbiamo avuto l'occasione di incontrare, già dal livello 2, la funzione essenziale utilizzata nel Basic Microsoft per manipolare le stringhe: si tratta di MID$ (A$, I J) che permette di estrarre, nella stringa A$, una sottostringa di lunghezza J, cominciante al primo carattere.
Così se,

A$ = "SIGNORE"

avremo:

MID$ (A$, 2, 3) = "IGN"

Per facilitare le manipolazioni, è possibile utilizzare le due istruzioni LEFT$ e RIGHT$;

LEFT$ (A$, I) permette di estrarre i I primi caratteri partendo dalla sinistra della parola A$.

Così LEFT$ (A$, 2) = "MO", con l'esempio precedente (potremmo d'altronde utilizzare MID$ (A$, 1, 2) per ottenere lo stesso risultato).

RIGHT$ (A$, J) permette di estrarre i J ultimi caratteri (si parte dalla destra della parola A$).

Così RIGHT$ (A$, 3) = "ORE", sempre con lo stesso esempio.
È più difficile fare a meno della funzione RIGHT$, ma sarebbe possibile fare MID$ (A$, LEN (A$)—3, 3) per ottenere lo stesso risultato.
Il Basic del Videopac non permette la sostituzione diretta di una sottostringa con un'assegnazione del tipo MID$ (—,—)= " ". Sarà necessario, per sostituire la I-esima lettera della stringa A$, scrivere:

A$ = LEFT$ (A$, I, 1) + "nuova lettera" + RIGHT$ (A$, LEN (A$) — I).

Delle formule analoghe potranno essere utilizzate per sostituire nelle A$ una sottostringa con un'altra, non necessariamente della stessa lunghezza, il che permette una grande agilità.

Ci pare utile riassumere le funzioni di trattamento delle stringhe o catene di caratteri, evidenziando le funzioni fra loro inverse:

–LEN (X$) = lunghezza della stringa A$;
–STR$ (X) converte un numero X in una stringa, è *l'inverso* di VAL (A$);
–VAL (X$) dà il valore numerico della stringa X$, è *l'inverso* di STR$ (X);

–ASC (X$) dà il codice ASCII del primo carattere di X$, è *l'inverso* di CTR$ (X);
–CHR$ (X) dà il carattere ASCII corrispondente a X, è *l'inverso* di ASC (X$).

VISUALIZZAZIONE A GRANDE FORMATO: TX(i,j,k)

Il Basic del VIDEOPAC+ dispone d'una possibilità assai rara che permette, in modo molto semplice, di migliorare l'aspetto grafico di certi giochi. Si può infatti, mediante un'istruzione facile da manipolare, ottenere dei caratteri espansi, sia in altezza che in larghezza, o entrambi contemporaneamente.

L'ISTRUZIONE TX(i,j,k)

I tre indici hanno i seguenti significati e debbono essere interi:

— (i) definisce il colore del primo piano, e varia da 0 a 7;
— (j) definisce la dimensione dei caratteri in positivo o in negativo;
— (k) abilita o no il lampeggio.

L'espansione dei caratteri implica però che più spazio dev'essere riservato sullo schermo e complica perciò leggermente la definizione degli spazi di visualizzazione sullo schermo. Basta però seguire il metodo seguente per risolvere facilmente il problema:

— *doppia altezza*, per riservare a ciascun carattere due caselle verticali si deve ripetere la stampa in due posizioni allineate l'una sotto l'altra;
— *doppia larghezza*, qui occorre riservare per ogni lettera due posizioni di stampa affiancate, per far ciò o si lascia uno spazio dopo ogni lettera o si ripete ciascuna lettera due volte;
— *doppia dimensione* (altezza e larghezza), si combinano le due utilizzazioni sopra descritte.

Le cose si complicano un po' allorché si debbono visualizzare dei risultati numerici.

CARATTERI SPECIALI GRAFICI

Il Basic del VIDEOPAC+ ha la possibilità di definire 96 caratteri speciali in modo testo ed altrettanti in quello grafico. Tenendo presente che ogni carattere è rappresentato da una matrice di (8 * 10) = 80 punti che devono essere rappresentati in *caratteri esadecimali*, quindi un totale di 20 valori, che seguono l'istruzione SETET (per il modo testo) o SETEG (per il modo grafico). Le corrispondenti istruzioni sono:

— ET (i,j,k) testo extra;
— EG (i,j,k) grafica extra;

in cui gli indici (i,j,k) hanno lo stesso significato che nella specifica TX. Per semplificare il vostro lavoro in tal senso, che vi consentirà d'arricchire e abbellire i vostri giochi, abbiamo inserito un programma d'utilità nell'ultima sezione che definisce automaticamente i codici, buon divertimento! Occorre tenere presente che i codici dei vostri caratteri speciali debbono essere compresi fra 32 e 127.

Un dado

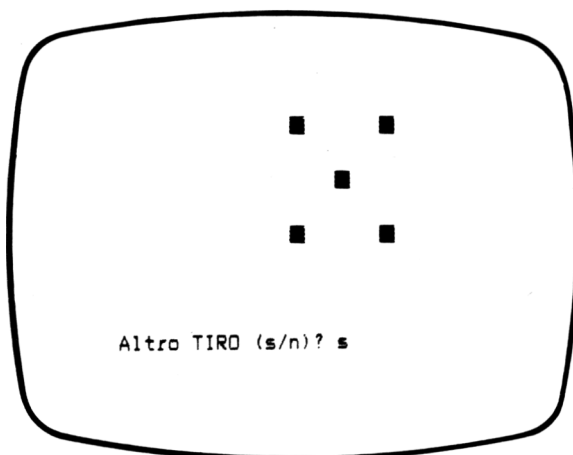
RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: lancio di un dado (con grafica).
Difficoltà del gioco: nessuna.

IL GIOCO: Il programma propone semplicemente di lanciare un dado aleatorio, e la sua rappresentazione grafica in grande formato. Potrà essere facilmente utilizzato come sottoprogramma, in ogni tipo di gioco che faccia richiamo al lancio di uno o più dadi.

IL PROGRAMMA: La linea 50 utilizza il generatore aleatorio del Videopac per tirare un numero da 1 a 6.

PER GIOCARE: Il ciclo J (70÷90) disegna i bordi del dado.
Le linee 125, 135, 145, 150, 160, 170, 180 sono usate per il disegno dei punti.
Le linee 190 e 200 permetteranno di rifare un lancio.

ESTENSIONI POSSIBILI: Si potrà introdurre una piccola musica che renderà più gradevole ogni lancio, fare sfilare i punti come se il dado rotolasse su una tavola, ecc.

```
00010 PRINT" -  Un  D A D O  -  041@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 A=INT(RND(1)*6+1)
00060 TX1,4,0:CURSORX 10:CURSORY 5
00065 PRINT"             "
00070 FOR J=0 TO 10
00080 CURSORX 10:CURSORY 6+J:PRINT" "
00085 CURSORX 22:CURSORY 6+J:PRINT" "
00090 NEXT J
00100 CURSORX 10:CURSORY 17
00105 PRINT"             "
00110 TX2,4,0:SOUND 2
00120 IFA=1THEN130
00125 CURSORX 13:CURSORY 8:PRINT" "
00130 IFA<4THEN140
00135 CURSORX 19:CURSORY 8:PRINT" "
00140 IFA<>6THEN155
00145 CURSORX 13:CURSORY 11:PRINT" "
00150 CURSORX 19:CURSORY 11:PRINT" "
00155 IFA/2=INT(A/2)THEN165
00160 CURSORX 16:CURSORY 11:PRINT" "
00165 IFA<4THEN175
00170 CURSORX 13:CURSORY 14:PRINT" "
00175 IFA=1THEN190
00180 CURSORX 19:CURSORY 14:PRINT" "
00185 TX4,4,0:CURSORY 20
00190 INPUT "Altro TIRO (s/n)";R$
00200 IFLEFT$(R$,1)="s" THEN40
```

Più grande, più piccolo

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: gioco di deduzione (un solo giocatore).
Difficoltà del gioco: difficile.

IL GIOCO:

Si ispira da un principio leggermente differente da quello del MASTERMIND; si tratta ugualmente di indovinare un numero (da 1 a 7 cifre), ma le indicazioni di cui si dispone sono questa volta da una parte il numero di cifre che, nel numero provato, sono superiori a quelle del numero da indovinare, e, d'altra parte, il numero di cifre che sono inferiori. Se, per esempio, il numero da indovinare è 3054, e che si provi 4251, la macchina risponderà 2 più grandi (sono il 4 e il 2), e 1 più piccolo (è l'1).

IL PROGRAMMA:

La linea 60 proibisce di chiedere dei numeri di più di 8 cifre.
Il numero da indovinare è posizionato, alla linea 80, in una variabile alfanumerica X$.
I paragoni col numero tentato sono fatti alla linea 135 (che testa se si ha trovato giusto), e nel ciclo I (150÷180), dove si fa il paragone "lettera per lettera". La visualizzazione del risultato è realizzata grazie alle linee 190 e 195.
Il contatore del numero di prove è alla linea 200.

PER GIOCARE:

Il lancio del programma con RUN provoca il messaggio Numero di CIFRE.
Si inserisce un numero inferiore a 8 e si batte RETURN. Il numero di tentativi appare, così come la domanda Num. PROVATO; si risponde inserendo un numero, seguito da RETURN. Il risultato è allora visualizzato nella maniera > e <.
Se non è esatto, si ricomincia (attenzione ad inserire unicamente dei numeri aventi altrettante cifre che il numero da indovinare). Se si trova, il Videopac scriverà AVETE TROVATO ———IN———PROVE. Poi UN'ALTRA PARTITA? (s/n); faremo s seguito da RETURN per cominciare una nuova partita.

ESTENSIONI POSSIBILI:

Potremo, come per il MASTERMIND, immaginare un gioco analogo con delle lettere, dei colori, o anche utilizzando degli altri simbolismi. Si può anche prevedere di far giocare il Videopac che avrà allora il ruolo di decodificatore.

```
00010 PRINT"MAGGIORE<>minore 042@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 INPUT "Numero di CIFRE :";C
00060 IFC>8GOTO 40
00070 E=1:F=1
00080 X$=STR$(INT(RND(1)*9 C+10 (C-E)))
00090 X$=RIGHT$(X$,LEN(X$)-1)
00100 CURSORX 1:CURSORY 5
00105 PRINT"PROVA Num.  NUM.provato ";
00110 PRINT"   >     <    ":PRINT
00115 FOR J=1 TO 37:CURSORX J:CURSORY 6:
00120 PRINTCHR$(96):NEXT J
00125 CURSORX1:CURSORYE+6:PRINT;TAB(4);
00130 PRINTE;TAB(13):INPUT P$
00135 IFP$=X$GOTO 220
00140 A=0:B=0
00150 FOR I=1 TO C
00160 V$=MID$(P$,I,1):W$=MID$(X$,I,1)
00165 IFV$>W$THENA=A+1
00170 IFV$<W$THENB=B+1
00180 NEXT I
00190 CURSORX 27:CURSORY E+6:PRINTA;
00195 PRINTTAB(6);B
00200 E=E+1:F=F+1:IFF=18THENF=1
00210 GOTO 125
00220 PRINT:PRINT"Hai trovato ";X$;
00225 PRINT"in ";E;" Prove":SOUND 4
00230 INPUT "Un'altra PARTITA(s/n):";R$
00240 IFLEFT$(R$,1)="s"THENGOTO 40
00250 PRINT"GRAZIE ed a Presto !":END
```

```
Numero di CIFRE :? 4

PROVA Num.  NUM.provato    >     <
.....................................
    1         ? 1234       1     3
    2         ? 5678       1     1
    3         ? 5555       1     2
    4         ? 5559       1     2
    5         ? 5549       1     2
    6         ? 5539       1     2
    7         ? 5521       0     2
    8         ? 5532       1     2
    9         ? 5526       1     2
   10         ? 5525       1     2
   11         ? 5518       1     2
```

Ordine

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco di riflessione.
Difficoltà del gioco: media.

IL GIOCO:

Le 10 prime lettere dell'alfabeto appaiono in disordine, bisogna riuscire a rimetterle in ordine, col minor numero possibile di tentativi, con delle permutazioni successive di due delle lettere; ma questo ha anche come inconveniente di far permutare attorno a lei le due lettere che la circondano, secondo il meccanismo spiegato qui sotto:

B A D G H J I C F E
0 1 2 3 4 5 6 7 8 9 : PERMUTAZIONE 83

B A H F D J I E G C
0 1 2 3 4 5 6 7 8 9 : RISULTATO

IL PROGRAMMA:

I cicli chiusi I (60÷120) e J (80÷100) definiscono una variabile alfanumerica A$, contenente le 10 prime lettere dell'alfabeto in disordine.
La linea 145 visualizza A$, e la linea 150 visualizza i numeri delle lettere da utilizzare per la permutazione.
Si inserisce la permutazione alle linee 190 e 195.
Le linee 200÷210 trasformano la variabile alfanumerica inserita in una variabile numerica.
La linea 220 chiama una prima volta il sottoprogramma d'inversione.
La linea 250 chiama una seconda volta il sottoprogramma d'inversione.
La linea 260 contabilizza i tentativi.
La linea 180 controlla i risultati.

PER GIOCARE:

Lanciare il programma con RUN. La parola iniziale appare, ogni lettera essendo riferita ad un numero da 0 a 9. Alla domanda INVERSIONE (I/J), si risponde inserendo un numero di due cifre xy; x è il numero della prima lettera, e y il numero della seconda. Le due inversioni sono allora successivamente effettuate, e si ricomincia. Il numero di tentativi apparirà alla fine della partita.

ESTENSIONI POSSIBILI:

Si può immaginare un gioco contro il Videopac, che valuterebbe il punteggio del giocatore in funzione delle sue proprie capacità; si possono anche modificare le regole che gestiscono le permutazioni.

```
00010 PRINT"METTI ORDINE 043@&"
00020 INPUT "DAMMI UN NUM.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 A$=""
00060 FOR I=1 TO 10
00070 R$=CHR$(INT(RND(1)*10+65))
00080 FOR J=1 TO I-1
00090 IFR$=A$(J)GOTO 70
00100 NEXT J
00110 A$=A$+R$
00120 NEXT I
00130 S=1
00140 CURSORX 1:CURSORY 4
00145 PRINT"RISULTATO :";A$
00150 PRINTTAB(11);"0123456789"
00160 CURSORX 1:CURSORY 9
00165 PRINT"PROVA Numero ";S
00170 CURSORX28:CURSORY12:PRINT"          "
00180 IFA$="ABCDEFGHIJ"GOTO300
00190 CURSORX1:CURSORY12:PRINT"INVERS";
00195 INPUT "IONE (I e J come IJ)";R$
00200 I=ASC(LEFT$(R$,1))-47
00210 J=ASC(RIGHT$(R$,1))-47
00220 GOSUB 500
00230 IFLEFT$(R$,1)="0"GOTO 250
00235 IFLEFT$(R$,1)="9"GOTO 250
00240 I=I-1:J=I+2
00250 GOSUB 500
00260 S=S+1
00270 GOTO 140
00300 PRINT"Trovato in ";S;" COLPI"
00310 SOUND5:SOUND0:SOUND2:SOUND4:END
00500 B$=""
00510 FOR T=1 TO LEN(A$)
00520 IFT=ITHENN$=MID$(A$,T,1)
00530 IFT=JTHENM$=MID$(A$,T,1)
00540 NEXT T
00550 FOR T=1 TO LEN(A$)
00560 IFT=ITHENB$=B$+M$:GOTO 590
00570 IFT=JTHENB$=B$+N$:GOTO 590
00580 B$=B$+MID$(A$,T,1)
00590 NEXT T
00600 A$=B$:RETURN
```

```
RISULTATO :IAACEHDHBD
           0123456789


PROVA Numero  1

INVERSIONE (I e J come IJ)? 37
```

L'impiccato

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco di lettere (versione a 2 giocatori).
Difficoltà del gioco: media.

IL GIOCO:

Sarebbe senz'altro difficile ritrovare il nome dell'inventore di questo gioco molto di moda, ma è certo che esiste da tempi immemorabili, qualche volta praticato sotto forme assai differenti (per esempio, ricostituire un maialino lanciando dei dadi).
L'apparizione dei giochi su microcalcolatore e la magia degli schermi video gli hanno fatto conoscere un rinnovo sicuro: in questa versione, uno dei giocatori deve inserire una parola di lunghezza qualunque. Il calcolatore visualizzerà allora dei puntini nella posizione delle lettere di quella parola e chiede, al secondo giocatore, di proporre una lettera.
Ogni risposta corretta fa apparire la lettera nella sua posizione esatta nella parola, ed ogni risposta sbagliata fa progredire il disegno del patibolo.
Per vincere, basta riuscire a trovare la parola sconosciuta prima di essere impiccato.

IL PROGRAMMA:

La parola da indovinare è introdotta all'inizio del programma (linea 30); una variabile B$, alla linea 40, contiene tutti gli elementi del disegno.
Il ciclo I (60-80) fabbrica una parola P$ che contiene i puntini nella posizione di ognuna delle lettere della parola iniziale.
Il ciclo I (120-140) si occupa del disegno (più o meno completo, secondo il valore di P). Le linee 150-160 visualizzano il risultato e arrestano il gioco.
La lettera provata è inserita alla linea 175 e testata in un ciclo I (190-210); il sottoprogramma 300-320 rimpiazza ogni puntino con la lettera corrispondente.

PER GIOCARE:

Dopo il lancio del programma con RUN, uno dei giocatori deve inserire la sua parola e battere RETURN. Dei puntini appaiono nella posizione di ognuna delle lettere della parola; e la domanda DAMMI LA TUA LETTERA appare sullo schermo; l'altro giocatore risponde allora proponendo una lettera seguita da RETURN.

Se la lettera è giusta, rimpiazzerà uno o più trattini; altrimenti, il patibolo comincerà a disegnarsi. La parola esatta sarà visualizzata in caso di perdita.

```
00010 PRINT"L' IMPICCATO 044@&"
00020 PRINT"Batti LOCK ??":SOUND4:PRINT
00030 INPUT "Dammi la tua PAROLA :";R$
00040 B$="z uj 5999ffffff 5  5f9 ///  "
00050 P$=""
00060 FOR I=1 TO LEN(R$)
00070 P$=P$+"-"
00080 NEXT I
00090 P=11
00100 INIT 7
00110 CURSORX 1:CURSORY 3:PRINTP$
00120 FOR I=11 TO PSTEP-1
00130 CURSORX 20:CURSORY 12-P:GR0,7,0
00135 PRINTMID$(B$,1+3*(I-1),3)
00140 NEXT I:TX0,0,0:PRINT:PRINT
00150 IFP$=R$THENPRINT"BRAVO!!":GOTO250
00160 IFP=1THENPRINT"PERSO ; ";:GOTO260
00170 CURSORX 1:CURSORY 18
00175 INPUT"Dammi la tua LETTERA :";S$
00180 T=0
00190 FOR I=1 TO LEN(R$)
00200 IFMID$(R$,I,1)=S$THENGOSUB 300
00210 NEXT I
00220 IFT<>0GOTO 110
00230 P=P-1
00240 GOTO 110
00250 SOUND4:SOUND5:SOUND5:SOUND4:END
00260 PRINT"la PAROLA era :";R$:SOUND 2
00270 SOUND2:SOUND5:SOUND4:SOUND0:END
00300 T=1
00310 B$=LEFT$(P$,I-1)+S$
00315 P$=B$+RIGHT$(P$,LEN(P$)-I)
00320 RETURN
```

PERSO ; la PAROLA era :FILM

Dammi la tua LETTERA :? D

Briscola

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco con le carte.
Difficoltà del gioco: facile.

IL GIOCO:

Il gioco proposto è piuttosto una simulazione, realizzata con 32 carte, della distribuzione di 4 mazzetti di 5 carte. Potrà essere utilizzato per lo studio di problemi facenti intervenire delle distribuzioni di 20 carte su 32.

IL PROGRAMMA:

Alla linea 50 si crea una tabella che può contenere le 20 carte.
Il ciclo I (90÷120) assicura la presa aleatoria di 20 numeri, presi da 32; il ciclo J (70-90) verifica che non si realizzi due volte la stessa presa.
Il secondo ciclo I (110-300) assicura in seguito la visualizzazione, il nome delle carte essendo scritto in chiaro, grazie alle linee 160, 170, 180 e 190.

PER GIOCARE:

Basta battere RUN per ottenere immediatamente l'apparizione di 4 mazzetti di 5 carte ognuno (una mano).
Un nuovo RUN permette di ottenere una nuova distribuzione.

ESTENSIONI POSSIBILI:

Si potrà facilmente mettere a punto un grafismo migliore utilizzando i caratteri grafici del Videopac; si potrà ugualmente servirsi di questo programma come di un nucleo per la messa appunto di programmi di giochi di carte su calcolatore (che potrà, se necessario, servire da partner: vedere l'esempio del POKER, nella stessa opera).

```
00010 PRINT"La BELLOTTA 045@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 DIM A(20)
00060 K=1
00070 CURSORX 8:CURSORY 12
00075 PRINT"Aspettate, sto TIRANDO !!!"
00080 FOR I=1 TO 20
00090 A(I)=1+INT(RND(1)*32)
00100 IFI=1 THEN140
00110 FOR J=1 TO I-1
00120 IFA(I)=A(J)THENJ=I-1:NEXTJ:GOTO90
00130 NEXT J
00140 NEXT I
00150 INIT 7
00160 FOR I=1 TO 20
00170 S$="Qu"
00180 S=1+A(I)-8*INT(A(I)/8)
00190 R$=STR$(S+6)+"  "
00195 R$=RIGHT$(R$,LEN(R$)-1)
00200 R$=LEFT$(R$,2)
00210 IFS=5THENR$="FA"
00220 IFS=6THENR$="DO"
00230 IFS=7THENR$="RE"
00240 IFS=8THENR$="AS"
00250 IFA(I)<=24THENS$="Fi"
00260 IFA(I)<=16THENS$="Pi"
00270 IFA(I)<=8THENS$="Cu"
00280 CURSORX8*INT((I+4)/5):CURSORY2*K
00290 PRINTR$;" ";S$;" "
00300 K=K+1
00310 IFK=6THENK=1
00320 NEXT I
```

```
RE Fi   7  Fi   FA Cu   7  Qu

8  Qu   RE Cu   FA Qu   10 Pi

8  Cu   9  Cu   9  Fi   9  Qu

9  Pi   DO Pi   DO Cu   8  Pi

AS Qu   FA Fi   8  Fi   10 Qu
```

Puzzle di calcolo

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: difficile.

IL GIOCO:

Si pratica fra due giocatori, uno ha il ruolo di codificatore, l'altro quello di decodificatore. Il codificatore parte da un numero (proposto dal Videopac) deve trasformarlo mediante 5 operazioni aritmetiche, che sceglie lui stesso come pure gli operandi.
I risultati ottenuti e le operazioni utilizzate vengono visualizzati sullo schermo: il decodificatore deve allora tentare di risalire al numero di partenza effettuando le operazioni in senso inverso.

IL PROGRAMMA:

La linea 50 consente d'ottenere il numero (intero) iniziale compreso fra 5 e 20.
Un ciclo I (110-200) permette d'introdurre successivamente le 5 operazioni.
4 sottoprogrammi sono utilizzati per i calcoli alle linee (1000-4000).
Dopo ogni calcolo appare il risultato sullo schermo (soltanto il codificatore dovrà guardare lo schermo durante questa operazione) e il risultato definitivo rimarrà in evidenza sullo schermo, ad uso del decodificatore, alla fine dei calcoli (linea 200).
Le linee 210 e 215 visualizzano le 5 operazioni usate, insieme con gli operandi corrispondenti.

PER GIOCARE:

Si batte RUN per lanciare il programma. Si evidenzia il numero iniziale, seguito dalla domanda "INTRODUCI 5 OPERAZIONI"; si dovrà introdurle una dopo l'altra, sotto la forma, per esempio: + 4 seguito da RET.
Il risultato verrà visualizzato dopo ogni operazione d'introduzione. Alla fine, le operazioni introdotte sono tutte evidenziate su una stessa riga dello schermo, ed il messaggio "Su CORAGGIO!!" è diretto al decodificatore (non vi è controllo sulla decodifica).

ESTENSIONI POSSIBILI:

Si potrà, naturalmente, cercare di migliorare la parte di decodifica del programma introducendo dei controlli sul risultato. Si può anche pensare di fare partecipare il Videopac maggiormente, affidandogli il ruolo della codifica, ecc...

```
00010 PRINT"PUZZLE di Calcolo 046@&"
00020 INPUT "Dammi un NUM.intero :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 X=INT(RND(1)*16+5)
00060 PRINT"NUMERO Iniziale :";X
00070 PRINT"Introdurre 5 OPERAZIONI"
00080 PRINT"In forma di +5. /2. *4 ..."
00090 F=X
00100 DIM R$(5)
00110 FOR I=1 TO 5
00120 CURSORX 21:CURSORY 7:PRINT"        "
00130 CURSORX 1:CURSORY 7
00135 INPUT "La tua OPERAZIONE :";R$(I)
00140 A=ASC(LEFT$(R$(I),1))-41
00150 IFA<=0ORA=3ORA=5ORA>6GOTO 120
00160 IFA>3THENA=(A/2)+1
00165 V=VAL(RIGHT$(R$(I).LEN(R$(I))-1))
00170 ONAGOSUB 1000.2000.3000.4000
00180 CURSORX19:CURSORY10:PRINT"        "
00190 CURSORX 1:CURSORY 10
00200 PRINT"NUMERO Ottenuto :";F:NEXT I
00210 PRINT:PRINT"OPERAZIONI :";R$(4);
00220 PRINT" ";R$(1);" ";R$(5);" ";
00230 PRINTR$(3);" ";R$(2):PRINT:PRINT
00240 PRINT:PRINT"Su ... CORAGGIO !!!"
00250 SOUND5:SOUND4:SOUND0:SOUND2:END
01000 F=F*V:RETURN
02000 F=F+V:RETURN
03000 F=F-V:RETURN
04000 F=F/V:RETURN
```

```
NUMERO Iniziale : 8
Introdurre 5 OPERAZIONI
In forma di +5, /2, *4 ...


La tua OPERAZIONE :? /6


NUMERO Ottenuto : 6

OPERAZIONI :+24 *2 /6 -18 +14



Su ... CORAGGIO !!!
```

Impiccato inverso

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco sulle parole.
Difficoltà del gioco: facile.

IL GIOCO:

L'IMPICCATO è un gioco di lettere molto noto, presentato d'altronde in questo libro, nel quale si deve indovinare una parola nascosta dal calcolatore e dove ogni lettera non scoperta vi porta un po' più vicini al patibolo. Qui, è il Videopac che cercherà di indovinare la vostra parola, ma lui gioca un po' a caso e non possiede la vostra intuizione; ci vorranno dunque, in regola generale, molti tentativi per scoprire la soluzione. Per aumentare le sue possibilità, ci appelliamo a una formula che gli permette di pescare di più nelle vocali, poi nelle consonanti di uso comune, e quindi nelle consonanti più rare.

IL PROGRAMMA:

La linea 50 permette di registrare, in una variabile alfanumerica D$, le lettere dell'alfabeto, nell'ordine: vocali, consonanti usuali, altre consonanti. La parola da indovinare è introdotta alla linea 70 (infatti, basterebbe introdurre la sua lunghezza).
Il ciclo I (90÷120) stampa i puntini nelle posizioni delle lettere da trovare.
Il ciclo alla linea 140 serve unicamente a simulare il tempo di ricerca della lettera.
La formula, piazzata alla linea 150, permette di modificare leggermente la probabilità di pesca, in favore delle vocali e delle consonanti usuali.
La linea 170 provoca una nuova pesca, se il ciclo I (210÷275) ristampa i puntini e le lettere trovate.
La linea 280 incrementa il contatore di tentativi.
La linea 320 visualizza il numero di tentativi realizzati, alla fine della partita.

PER GIOCARE:

Lanciare il programma con RUN.
Inserire la parola da scoprire, seguita da RETURN. I puntini sono allora visualizzati (uno per lettera) e il Videopac propone una prima lettera, guarda se va bene e continua automaticamente, fino a che avrà trovato la parola. Si potrà uscire dal programma ad ogni momento, battendo RESTORE.

ESTENSIONI POSSIBILI: Sarà senz'altro interessante, per i bambini, di prevedere una versione non automatica dove il giocatore risponderebbe SI o NO per ogni lettera proposta dal Videopac; si potrà anche immaginare di aggiungere una visualizzazione grafica, per esempio un patibolo o una macchina che va verso un burrone, ecc.

```
00010 PRINT"IMPICCATO/Rovescio 047@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1TOF:Y=RND(1):NEXT I:PRINT
00040 CLEAR150:PRINT"Batti LOCK !??!"
00050 D$="YAEIOUBCDFGLMNPRSTVWHKJQXZ"
00060 C=0:T$="":PRINT:PRINT:PRINT
00070 INPUT "Dammi la tua PAROLA :";R$
00080 INIT 7
00090 FOR I=1 TO LEN(R$)
00100 CURSORX I:CURSORY 5:PRINT"-"
00110 T$=T$+"-"
00120 NEXT I
00130 CURSORX 1:CURSORY 9
00135 PRINT"Cerco....: PROVA ";C
00140 FOR J=1 TO 1000:NEXT J
00150 X=-6*(RND(1)<.7)-13*(RND(1)<.5)
00155 X=X-7*(RND(1)<.4)
00160 X=INT(X*RND(1)+1)
00170 IFMID$(D$,X,1)="-"GOTO 150
00180 E$=MID$(D$,X,1)
00190 B$=LEFT$(D$,X-1)+"-"
00195 D$=B$+RIGHT$(D$,26-X)
00200 PRINT:PRINT:PRINT"PRENDO :";E$
00205 SOUND 4:SOUND 4:SOUND 4:SOUND 4
00210 FOR I=1 TO LEN(R$)
00220 IFE$<>MID$(R$,I,1)GOTO 260
00230 Z$=LEFT$(T$,I-1)+E$
00240 T$=Z$+RIGHT$(T$,LEN(T$)-I)
00250 SOUND 0:SOUND 0:SOUND 0:SOUND 0
00260 CURSORX I:CURSORY 5
00270 PRINTMID$(T$,I,1)
00275 NEXT I
00280 C=C+1
00290 IFT$=R$GOTO 310
00300 GOTO 130
00310 CURSORX 1:CURSORY 18
00320 PRINT"Trovato in ";C;" COLPI"
00330 SOUND 4:SOUND 4:SOUND 4:END
```

```
IMPONDERABILITA

Cerco....: PROVA  21

PRENDO :R


Trovato in  22 COLPI
```

Nicomaco

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco di riflessione (calcolo mentale).
Difficoltà del gioco: media.

IL GIOCO:

Utilizza una proprietà un po' particolare degli interi da 1 a 100 che possono essere identificati senza ambiguità dai loro resti nelle divisioni per 3, 5 e 7.
Il programma, proposto qui, è un programma codificatore che visualizza semplicemente i resti di un intero dato N nelle divisioni indicate.
Il gioco consiste nel cercare di scoprire l'intero (ma, in questa versione, avrete diritto ad un solo tentativo).

IL PROGRAMMA:

La linea 50 permette di creare l'intero N.
Il ciclo I (80÷130) calcola i tre resti successivi, ed il ciclo chiuso J (100÷120) permette di farli apparire in una forma assai visuale, nelle giuste locazioni.
Le linee (60÷75) sono lì per indicare di quale divisione si tratta.
Il numero proposto dal giocatore è inserito alla linea 145; è visualizzato e testato alle linee 150 e 160 (messaggio di fine partita : BRAVO o PERSO).
Le linee 180 e 190 si occupano di gestire la prossima partita, secondo la risposta del giocatore.

PER GIOCARE:

Si lancia il programma battendo RUN. I tre resti saranno allora visualizzati sulla stessa linea, simbolizzati con degli asterischi (l'assenza di asterischi significante: resto 0). Alla domanda TUA PROVA, si risponderà con un intero tra 0 e 100, seguito da RETURN.
Si otterrà in risposta un messaggio PERSO o BRAVO, poi la domanda UN'ALTRA PARTITA? (s/n). Per fare un nuovo tentativo, si risponderà s, seguito da RETURN.

ESTENSIONI POSSIBILI:

Si può immaginare di realizzare il programma decodificatore (si danno i resti ed il Videopac dovrà darvi N) e cercare se la proprietà può essere estesa (effettuando altre divisioni, per esempio).

```
00010 PRINT"NICOMACO 048@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 X=INT(RND(1)*100+1)
00060 PRINT"RESTO della Divisione per"
00070 PRINT:PRINTTAB(12);"3";
00075 PRINT TAB(20);"5";TAB(28);"7"
00080 FOR I=1 TO 3
00090 A=X-(2*I+1)*INT(X/(2*I+1))
00100 FOR J=1 TO A
00110 CURSORX J+8*I+3:CURSORY5:PRINT"*"
00120 NEXT J
00130 NEXT I
00140 CURSORX 1:CURSORY 10
00145 INPUT "Tua PROVA :";N
00150 IFN=XTHENPRINT:PRINT"BRAVO!?!"
00160 IFN<>XTHENPRINT:PRINT"PERSO?!?"
00170 SOUND 0:SOUND 0:SOUND 0:PRINT
00180 INPUT "Un'altra PARTITA(s/n)";R$
00190 IFR$="s"THENPRINT:GOTO 20
```

```
RESTO della Divisione per

            3       5       7
           *       *       *****



Tua PROVA :? 66

PERSO?!?

Un'altra PARTITA(s/n)? s
```

Corsa di cavalli

RIFERIMENTI: Difficoltà di programmazione: media. Categoria del gioco: fortuna (vari giocatori). Difficoltà del gioco: nessuna.

IL GIOCO: Si inserirà un numero di cavalli compreso tra 1 e 10, e la corsa comincerà immediatamente. Ci potranno essere dei pareggi, poiché si terrà conto del fatto che il Videopac fa avanzare i cavalli gli uni dopo gli altri, cominciando col numero 1; la classifica sarà visualizzata man mano, secondo gli arrivi (e tenendo conto dei pareggi).

IL PROGRAMMA: Un primo ciclo I (100÷180) visualizza i cavalli nella loro posizione di partenza.
Un secondo ciclo I (100-160), utilizzante lo stesso NEXT I, li farà avanzare; la posizione di ogni cavallo sarà individuata con una cifra o una lettera in una variabile alfanumerica A$.
Le linee 125 e 130 faranno avanzare i cavalli.
La linee 140 gestisce gli arrivi.
Le linee 150 e 155 realizzano le stampe.
Il sottoprogramma (240÷280) fa avanzare il contatore di arrivi, e stampa i risultati.

PER GIOCARE: Si lancia il programma con RUN. Alla domanda: NUMERO CAVALLI?: si risponde con un intero da 1 a 10, seguito da RETURN. La corsa si svolgerà in seguito automaticamente.

ESTENSIONI POSSIBILI: Potremo disegnare i cavalli con un grafismo più curato, e differenziarli con il colore.
Si potrebbe anche immaginare un sistema di scommesse, che interverrà prima della partenza di ogni corsa.

```
00010 PRINT"Corsa di CAVALLI 0490&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 INPUT "Numero CAVALLI :";N
00055 IFN>9THEN40
00060 A$="":P=0:A=0:T=1:K=0
00070 J=T
00080 A$=A$+"0":I=J
00090 GOTO 150
00100 FOR I=1 TO N
00110 IFMID$(A$,I,1)="K"THEN180
00120 V$=MID$(A$,I,1)
00125 B$=CHR$(ASC(V$)+INT(RND(1)*4+1))
00130 C$=RIGHT$(A$,LEN(A$)-I)
00135 A$=LEFT$(A$,I-1)+B$+C$
00140 IFB$>="K"THENGOSUB 240
00150 CURSORX ASC(MID$(A$,I,1))-46
00155 CURSORY 2*I+3:PRINT"      ";I
00160 SOUND 4:FOR D=1 TO 600:NEXT D
00170 IFK=0THENJ=J+1:IFJ<=NGOTO 80
00175 IFK=0ANDJ>NTHENK=1:GOTO 100
00180 NEXT I
00190 P=P+T
00200 T=0
00210 IFA<NGOTO 100
00220 SOUND 2
00230 END
00240 U$=RIGHT$(A$,LEN(A$)-I)
00245 A$=LEFT$(A$,I-1)+"K"+U$
00250 CURSORX 1:CURSORY 2*I+3:PRINTP
00260 T=1:SOUND 2
00270 A=A+T
00280 RETURN
```

```
Numero CAVALLI :? 6

          1
        2
          3
       4
          5
         6
```

Attraverso

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessione.
Difficoltà del gioco: media.

IL GIOCO:

Sullo schermo appare un quadrato di 9 caselle, con, sul bordo verticale, delle lettere e, sul bordo orizzontale, delle cifre. Uno dei giocatori manipola le cifre, l'altro le lettere. Tre spostamenti sono possibili per le cifre, e tre altri per le lettere. Le lettere possono spostarsi verso l'alto, a destra e a sinistra, e le cifre verso la destra, in alto od in basso.
Il vincitore è il primo che riesce a fare uscire tutti i suoi simboli (le lettere escono in alto, le cifre a destra).
Non ci sono prese, e i salti non sono autorizzati; non si può neanche posizionarsi su una casella già occupata.

IL PROGRAMMA:

Le posizioni delle cifre e delle lettere sono registrate in una tabella N (4,4); le linee da 10 a 70 definiscono i valori di partenza.
Le linee da 90 a 140 definiscono il tavolo di gioco e le linee 150 e 155 visualizzano il giocatore in gioco (in maniera semplificata 1 per le cifre e A per le lettere).
Le linee 190 e 195 controllano il colpo giocato, così come i cicli I e J (200÷240) ed il cambiamento di giocatore è assicurato dalla linea 330.

PER GIOCARE:

Il tavolo di gioco apparirà dopo aver battuto RUN, e il calcolatore aspetta la mossa del giocatore individuato con un 1 o una A.
Si inserirà prima la cifra o la lettera rappresentante la pedina da spostare, seguita da una indicazione di direzione nella forma N (per Nord), S (per Sud), E (per Est), O (per Ovest).
Si potrà così fare 2E seguito da RETURN. Se la mossa non è valida, non sarà eseguita, ed il codice del giocatore rimarrà alla visualizzazione; se è valida, lo spostamento sarà effettuato, e si otterrà il messaggio corrispondente al secondo giocatore. Il primo che riuscirà a far sparire le sue pedine avrà vinto.

ESTENSIONI POSSIBILI: Si può immaginare di estendere il gioco con più cifre e lettere e di migliorare i controlli. Si può anche definire una strategia per il Videopac.

```
00010 PRINT"A T T R A V E R S 0 050@&"
00020 DIMN(4,4):K=1
00030 FOR I=1 TO 4
00040 N(I,1)=ASC("0")+I
00050 N(4,I)=ASC("?")+I
00060 NEXT I
00070 N(4,1)=0
00080 INIT 7:PAGE:STORE
00090 FOR U=1 TO 4
00100 FOR V=1 TO 4
00115 CURSORX 4*U:CURSORY 4*V-2:
00120 IFN(V,U)=0THENPRINT("."):GOTO 130
00125 PRINTCHR$(N(V,U))
00130 NEXT V
00140 NEXT U
00150 STORE:CURSORY17:PRINT"Gioca";
00155 PRINT"tore con ";:TX5,4,0
00160 IFK=0THENPRINT"LETTERE":GOTO170
00165 PRINT"CIFRE"
00170 TX5,0,1:PRINT" (Batti LOCK !!??)"
00175 CURSORY 20:SCREEN:DISPLAY:TX0,0,0
00180 INPUT "Tuo TIRO (N,S,E,O) :";R$
00185 Y$=RIGHT$(R$,1)
00190 IFK=1ANDY$="O"GOTO 175
00195 IFK=0ANDY$="S"GOTO 175
00200 FOR I=1 TO 4
00210 FOR J=1 TO 4
00220 IFN(I,J)=ASC(LEFT$(R$,1)) THEN260
00230 NEXT J
00240 NEXT I
00250 GOTO 150
00260 A=I:B=J
00270 I=I-(Y$="S")+(Y$="N")
00280 J=J-(Y$="E")+(Y$="O")
00290 IF(I=0ANDK=0)OR(J>4ANDK=1)GOTO320
00300 IF N(I,J)=0THEN310
00305 J=4:NEXT J:I=4:NEXT I:GOTO 150
00310 N(I,J)=ASC(LEFT$(R$,1))
00320 J=4:NEXT J:I=4:NEXT I:N(A,B)=0
00330 K=ABS(K-1)
00340 GOTO 80
```

```
1   .   .   .

.   2   .   .

3   .   .   C

.   A   B   .

Giocatore con CIFRE
 (Batti LOCK !!??)

Tuo TIRO (N,S,E,O) :? 3E
```

Mastermind

RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: logica (un solo giocatore).
Difficoltà del gioco: difficile.

IL GIOCO: Diventato un grande classico dei giochi su microcalcolatore, questo gioco sviluppa il ragionamento deduttivo; bisogna riuscire a trovare il più velocemente possibile un numero di N cifre (potremo prendere N compreso tra 1 e 5), disponendo ad ogni tentativo di informazioni. In effetti, il Videopac risponderà indicando da una parte il numero di cifre che sono, nel numero proposto, esattamente nella stessa posizione che nel numero da scoprire, e, d'altra parte, il numero di cifre che figurano nel numero da scoprire senza essere esattamente nel posto giusto.

IL PROGRAMMA: La scelta del numero di cifre del numero da scoprire è effettuata alle linee 40 e 50. Il numero è creato alla linea 60 (potranno esserci delle cifre ripetute).
La visualizzazione è preparata dalle linee 90 e 95.
Le linea 130 e 135 si occupano della fine della partita. Un primo ciclo I (140÷230) testa le cifre che sono nel posto giusto.
Un secondo ciclo J (160÷180) testa le cifre che sono presenti e mal posizionate.
I contatori sono alle linee 200 e 220.
Le linee 240 e 250 visualizzano il risultato, e la linea 260 contabilizza i tentativi.

PER GIOCARE: Si batte RUN e si sceglie la lunghezza del numero desiderato (da 1 a 6), seguita da RETURN. Dopo il messaggio N. NUMERO GIUSTO POSTO G. , si introduce il proprio tentativo, seguito da RETURN. Si ottiene immediatamente la risposta (in caso di doppia cifra, i risultati possono essere un po' delicati da interpretare).
Si introduce in seguito un nuovo tentativo, e così via. Quando si trova, si otterrà il messaggio TROVATO IN ——— PROVE.
Si può uscire dal programma in ogni momento con RESTORE.

ESTENSIONI POSSIBILI: Si può cercare di evitare la presenza di cifre doppie nel numero da trovare, si può anche prevedere un cambiamento automatico di pa-

gina, un richiamo dei risultati precedenti; potremo anche realizzare dei giochi analoghi con delle parole, o altri simboli (colori, forme...).

```
00010 PRINT"MASTERMIND 051@&"
00020 INPUT "Dammi un NUM.intero :";F
00030 FOR I=1 TO F;Y=RND(1):NEXT I
00040 INPUT "Quante CIFRE (<=6) :";N
00050 IFN>6GOTO 40
00060 E=1:F=1:PAGE
00065 X$=STR$(INT(RND(1)*9 N+10 (N-1)))
00070 X$=RIGHT$(X$,LEN(X$)-1)
00080 INIT 7:STORE
00090 PRINT"PROVA     NUMERO     ";
00095 PRINT"CIFRA     POSTO"
00100 FORI=1TO39:PRINTCHR$(126);:NEXT I
00105 SCREEN:DISPLAY:PRINT:GOTO115
00110 CURSORX 1:CURSORY 3+F
00115 PRINTE;TAB(8):INPUT E$
00120 X=0:Y=X
00130 IFE$<>X$GOTO 140
00135 PRINT"Trovato in ";E;" COLPI":END
00140 FOR I=1 TO N
00145 Z$=MID$(X$,I,1):W$=MID$(E$,I,1)
00150 IFZ$=W$ GOTO 220
00160 FOR J=1 TO N
00165 H$=MID$(X$,I,1):Z$=MID$(E$,J,1)
00170 IFJ<>IANDH$=Z$GOTO 200
00180 NEXT J
00190 GOTO 230
00200 Y=Y+1
00210 GOTO 230
00220 X=X+1
00230 NEXT I
00240 CURSORX 23:CURSORY 3+F
00250 PRINTX;TAB(10):PRINTY
00260 E=E+1:F=F+1
00270 IFF=18THENF=1
00280 GOTO 110
```

```
PROVA      NUMERO      CIFRA      POSTO
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 1        ? 01          0          1
 2        ? 12          0          0
 3        ? 20          1          0
 4        ? 10          1          0
 5        ? 30          1          0
 6        ? 40
Trovato in  6 COLPI
```

Idem

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco di calcolo mentale (2 giocatori).
Difficoltà del gioco: secondo il livello scelto.

IL GIOCO:

Il Videopac sceglie un numero N di cifre (secondo il livello di difficoltà indicato all'inizio). Questo numero (lo stesso all'inizio per i 2 giocatori) sarà regolarmente decrementato ad ogni giro, di una quantità ugualmente scelta a caso dal Videopac all'inizio del gioco; voi dovete, ad ogni colpo, scegliere una colonna (le decine, le centinaia, ecc.) ed aggiungere in questa colonna un certo intero; lo scopo è di riuscire ad ottenere ad un certo momento la visualizzazione di un numero avente tutte le cifre identiche. Ogni risultato sarà visualizzato dopo un breve istante, dopodiché la mano passerà al giocatore seguente, che ripartirà dal suo risultato precedente.

IL PROGRAMMA:

Una variabile numerica N è dimensionata a 2 alla linea 40, per contenere i risultati successivi di due giocatori.
La linea 50 permetterà di inserire il livello di difficoltà scelto.
Il numero iniziale è creato dalla linea 70. La quantità che interverrà nel decremento è ottenuta alla linea 90. La linea 120 testa se il risultato non è troppo piccolo.
La linea 120 visualizza il numero del giocatore.
Il numero è visualizzato alle linee 140 e 145 e il gioco del giocatore è inserito alle linee 160 e 165 (numero di due cifre).
Le linee 170 e 175 incrementano la colonna scelta, e la linea 180 decrementa il numero della quantità scelta in partenza.
Il risultato sarà visualizzato un breve istante (linee 210 e 215). Il ciclo I (260÷280) permette di verificare se tutte le cifre sono identiche. La stampa e la fine della partita sono effettuate alla linea 290. Le linee 300 e 310 si occupano del cambiamento del giocatore.

PER GIOCARE:

Lanciare l'esecuzione del programma con RUN. Il numero iniziale sarà visualizzato per il giocatore N. 1, che gioca battendo xy, seguito da RETURN; x è la cifra da aggiungere, e y è il numero della colonna dove bisogna aggiungerla.

Il risultato (provvisorio se è troppo piccolo) sarà allora visualizzato un breve istante, e la linea passa al secondo giocatore, e così via. Si può fermare in ogni momento il programma battendo RESTORE.

ESTENSIONI POSSIBILI: Si potrà estendere il gioco a più di 2 giocatori, migliorare la scelta del numero intervenente nella decrementazione, migliorare i diversi controlli, lasciare i differenti risultati visualizzati, ecc.

```
00170 M=U (VAL(RIGHT$(R$,1))-1)
00175 N(H)=N(H)+VAL(LEFT$(R$,1))*M
00180 N(H)=N(H)-A
00190 IFN(H)>U NTHENN(H)=N(H)-U N
00200 N(H)=INT(N(H))
00210 CURSORX 1:CURSORY 13
00215 PRINT"RISULTATO :";N(H)
00220 SOUND 5:SOUND 5:SOUND 4
00230 FOR J=1 TO 1200:NEXT J
00240 R$=STR$(N(H))
00250 R$=RIGHT$(R$,LEN(R$)-1)
00260 FOR I=1 TO LEN(R$)-1
00265 S$(I)=MID$(R$,I,1)
00270 Z$(I)=MID$(R$,I+1,1)
00275 IFS$(I)<>Z$(I)GOTO 300
00280 NEXT I
00290 SOUND 2:SOUND 2:SOUND 0:END
00300 H=H+1
00310 IFH>=3THENH=1
00320 GOTO 100
```

```
00010 PRINT"-I D E M- 052@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:H=1:DIM N(2)
00050 INPUT "LIVELLO (2-7) :";N
00060 U=10:P=0
00070 M=(U N-U (N-1))*RND(1)+U (N-1)
00080 N(1)=INT(M):N(2)=N(1)
00090 A=INT(2*U (N-1)*RND(1)+U (N-1))
00100 IFH=1THENP=P+1
00110 INIT 7
00120 IFN(H)>=U (N-1)GOTO 130
00125 N(H)=U N-ABS(N(H))
00130 CURSORX 1:CURSORY 3:PRINT"GIOCA";
00135 PRINT"TORE :";H;"   PROVA :";P
00140 CURSORX 1:CURSORY 6:PRINT"NUMERO"
00145 CURSORX 15-N:CURSORY 6:PRINTN(H)
00150 CURSORX9:CURSORY8:PRINT"7654321 "
00160 CURSORX1:CURSORY10:PRINT"Il tuo";
00165 INPUT" GIOCO (N e C come NC) ";R$
```

```
GIOCATORE 1        PROVA 2

NUMERO:        903
          7654321

IL TUO GIOCO (N e C come NC)
21

RISULTATO: 776
```

Orologio

RIFERIMENTI:
Difficoltà di programmazione: difficile.
Categoria del gioco: disegno.
Difficoltà del gioco: nessuna.

IL GIOCO:
Trasformate il vostro televisore in orologio numerico a visualizzazione in grande formato, grazie a questo programma che vi permetterà di animare una vetrina, di fare un orologio di fantasia, e che potrà prestarsi a molti altri usi.

IL PROGRAMMA:
La linea 20 vi permetterà di inserire l'ora sotto forma di un intero di 6 cifre (per esempio: 102436 significa ore 10, 24 minuti e 36 secondi).
Le linee 80÷160 realizzano il disegno, scalando la stampa man mano, con l'aiuto della variabile K, incrementata alla linea 165.
Le linee da 235 a 290 assicurano la visualizzazione dei secondi ed il cambiamento dell'ora.

PER GIOCARE:
Subito dopo il lancio con RUN, il programma si mette in attesa dell'ora; si batte al prossimo top sonoro e si aspetta il top per battere RETURN. L'ora appare allora molto rapidamente e l'orologio è in marcia. Per fermarlo, bisognerà battere RESTORE.
L'ora sarà inserita sotto forma di un intero avente imperativamente 6 cifre; per le ore 5 e 6 minuti, si dovrà battere 050600 seguito da RETURN.

ESTENSIONI POSSIBILI:
Sarà interessante modificare il programma per fare una sveglia, con l'allarme; si può anche migliorare la grafica, introdurre il colore.

```
00010 PRINT" OROLOGIO-1 053@&"
00020 INPUT "ORA (hhmm):";A$
00030 INIT 7:TX0,1,0:CURSORX27:CURSORY8
00035 STORE:PRINT"OROLOGIO-1":CURSORX27
00040 CURSORY 9:PRINT"OROLOGIO-1":K=1
00045 SCREEN:DISPLAY:H$=CHR$(127)
00050 A=VAL(MID$(A$,K,1)):STORE
00060 TX0,0,0:IFA<>1ANDA<>4GOTO70
00065 GOTO 80
00070 CURSORX 6*K-5:CURSORY 1
00075 PRINTH$+H$+H$
00080 FOR J=1 TO 5
00090 IFA=1ORA=2ORA=3ORA=7GOTO 100
00095 CURSORX 6*K-5:CURSORY J:PRINTH$
00100 IFA=5ORA=6GOTO110
00105 CURSORX 6*K-3:CURSORY J:PRINTH$
00110 IFA=0ORA=2ORA=6ORA=8GOTO310
00120 IFA=2GOTO 130
00125 CURSORX 6*K-3:CURSORY J+4:PRINTH$
00130 NEXT J
00140 IFA=0ORA=1GOTO 150
00145 CURSORX6*K-4:CURSORY5:PRINTH$+H$
00150 IFA=1ORA=4ORA=7GOTO 165
00155 CURSORX 6*K-5:CURSORY 9
00160 PRINTH$+H$+H$:SCREEN:DISPLAY
00165 K=K+1
00170 IFK<5GOTO 50
00180 A=VAL(A$)+1
00190 IFA=2360THENA=0
00195 A$=RIGHT$(STR$(A),LEN(STR$(A))-1)
00200 A$="000"+A$
00210 A$=RIGHT$(A$,LEN(A$)-3)
00215 IFRIGHT$(A$,2)="60"THENGOSUB 330
00235 H=1
00240 FOR J=1 TO 60
00250 FOR I=1 TO 6
00260 CURSORX11:CURSORY5:IFH=1GOTO 270
00265 PRINTCHR$(34):GOTO 275
00270 GR0,7,0:PRINTCHR$(17)
00275 NEXT I:SOUND 4
00280 H=ABS(H-1)
00290 NEXT J
00300 SOUND4:SOUND6:GOTO 30
00310 CURSORX 6*K-5:CURSORY J+4:PRINTH$
00320 GOTO 120
00330 A$=STR$(VAL(LEFT$(A$,2))+1)
00340 A$=RIGHT$(A$,2)+"00":RETURN
```

Regine sulla scacchiera

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: difficile.

IL GIOCO:

Il gioco utilizza un algoritmo molto prestante per risolvere un problema che ha dato filo da torcere a dei ricercatori, anche celebri come il matematico GAUSS, che non è riuscito a scoprire tutte le soluzioni. Si tratta di trovare tutte le maniere di disporre delle regine su una scacchiera, in modo tale che nessuna possa mangiarne un'altra. Il programma è molto elastico, visto che disponete anche della possibilità di scegliere la dimensione della scacchiera, e i risultati saranno visualizzati in forma sia grafica che numerica.

IL PROGRAMMA:

La linea 30 vi permette di scegliere la dimensione della scacchiera.
Una variabile D sarà dimensionata, di conseguenza, alla linea 40.
Un ciclo di esplorazione principale I (50-110) contiene un sottociclo J (80÷100) che realizza il controllo di non allineamento (linee 90 e 95).
Se, nel corso dell'esplorazione, si trova un allineamento, si torna indietro, utilizzando le linee da 230 a 270 e si ricomincia.
Il ciclo F (130÷210) garantisce la stampa della scacchiera, ed il ciclo K (160-180), che potremo eventualmente sopprimere, assicura la stampa sotto forma di un codice numerico corrispondente alla posizione delle pedine in ogni colonna.
La linea 225 permette un arresto dopo l'apparizione di ognuna delle soluzioni.

PER GIOCARE:

Si lancia il programma con RUN, poi si battono le dimensioni della scacchiera desiderata, e si fa RETURN.
Basta aspettare, con pazienza (più la scacchiera sarà grande, più l'attesa sarà lunga). La prima soluzione apparirà, poi il programma si ferma, e bisogna battere il tasto RET per farlo ripartire. Per terminare prima dell'esplorazione di tutte le soluzioni, si batterà RESTORE.

ESTENSIONI POSSIBILI:

Si può immaginare di trovare degli algoritmi per tutti i problemi analoghi (posizioni di torri, spostamento di un cavaliere, ecc.).

```
00010 PRINT"REGINE su Scacchiera 054@&
00020 INIT 7:PAGE
00030 INPUT "DIMENSIONE Scacchiera :";S
00040 DIM D(S):TX0,0,1:PRINT:PRINT
00050 I=1:PRINT"Un po' di PAZIENZA !!!"
00060 D(I)=1
00070 IFI=1THEN110
00080 FOR J=1 TO I-1
00090 IFD(I)=D(J)THEN240
00095 IFABS(D(I)-D(J))=I-JTHEN240
00100 NEXT J
00110 I=I+1:IF I<=S THEN60
00120 INIT 7:STORE
00130 FOR F=1 TO S
00140 CURSORX 2*(F+1):CURSORY 0:PRINTF
00150 CURSORY 2*F:PRINTF
00160 FOR K=1 TO S
00170 CURSORX2*K+3:CURSORY2*F:PRINT"."
00180 NEXT K
00190 CURSORX2*D(F)+3:CURSORY2*F
00195 PRINT"*":CURSORY19:PRINT"Regine :"
00200 CURSORX F:CURSORY20
00205 PRINTRIGHT$(STR$(D(F)),1),
00210 NEXT F:SCREEN:DISPLAY
00220 SOUND 2:CURSORY 20:TX0.0,1
00225 INPUT "Con RET continui";R$
00230 I=I-1:GOTO 245
00240 J=I-1:NEXT J
00245 D(I)=D(I)+1
00250 IFD(I)<=STHEN70
00260 I=I-1
00270 IFI<>0THEN245
```

```
 1  * . . . . . . .

 2  . . . . * . . .

 3  . . . . . . . *

 4  . . . . . * . .

 5  . . * . . . . .

 6  . . . . . . * .

 7  . * . . . . . .

 8  . . . * . . . .


Regine :
15863724               Con RET continui?
```

Ripetizioni

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: gioco di memoria (uno o più giocatori).
Difficoltà del gioco: media.

IL GIOCO:

Dei numeri appariranno uno ad uno. Il giocatore dovrà ad ogni mano ripeterli tutti, dall'inizio (il che significa che ne avrà uno in più da battere ogni volta). Dovrà cercare di resistere il più possibile. Il miglior punteggio è memorizzato dopo ogni partita, il che permetterà eventualmente di giocare in molti.

IL PROGRAMMA:

La linea 50 chiama un sottoprogramma (linee da 1000 a 1060) che dà la regola del gioco.
La linea 70 crea le cifre, che sono immagazzinate in una variabile alfanumerica A$ (linea 80).
La linea 90 permette la stampa dell'ultimo numero a caso sulla quarta linea dello schermo, e la durata della stampa è determinata dalla linea 100.
La linea 120 aspetta la risposta del giocatore, che è paragonata alla risposta corretta alla linea 130.
La linea 160 marca i punti.
La linea 170 controlla il miglior punteggio.
I messaggi di fine partita sono effettuati dalla linea 200 alla linea 240.

PER GIOCARE:

Il lancio del programma con RUN provoca la visualizzazione della regola del gioco, poi di una cifra che scompare. L'apparecchio pone allora la domanda LA TUA SEQUENZA? Gli si risponde ridando la cifra, seguita da RETURN. Se è giusto, si ha per un breve istante il messaggio ESATTO. ATTENZIONE, poi una nuova cifra appare, scompare, e ancora la domanda LA TUA SEQUENZA? Si ribattono allora le due cifre già ottenute, seguite da RETURN.
Si continua così (3 cifre, poi 4, 5, ecc.) fino a che si commetta un errore, il che provoca l'apparizione del punteggio (e del miglior punteggio ottenuto fino ad allora), seguito dalla domanda UN'ALTRA PARTITA? (s/n); si batte s per continuare.
Si può anche lasciare il programma in ogni istante, battendo RESTORE prima dell'apparizione di una cifra.

ESTENSIONI POSSIBILI: Si potranno realizzare dei giochi analoghi e più o meno perfezionati rimpiazzando le cifre con delle lettere, o dei colori, o dei suoni...

```
00010 PRINT"RIPETIZIONI 055@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 M=0:A$=""
00050 GOSUB 1000
00060 H=0
00070 X=INT(RND(1)*10)
00080 A$=A$+RIGHT$(STR$(X),1)
00090 CURSORXRND(1)*30:CURSORY8:PRINT X
00100 FOR I=1 TO 200+RND(1)*300:NEXT I
00110 INIT 7:CURSORX 1:CURSORY 10
00120 INPUT "La tua SEQUENZA :";R$
00130 IFR$<>A$GOTO 190
00140 PRINT:SOUND 4:SOUND 2:SOUND 0
00145 PRINT"ESATTO - ATTENZIONE !!!"
00150 FOR I=1 TO 400:NEXT I
00160 H=H+1
00170 IFM<HTHENM=H
00180 GOTO 70
00190 INIT 7
00200 PRINT"Il tuo PUNTEGGIO e' :";H
00210 PRINT:PRINT"MIGLIOR Punteggio:";M
00220 PRINT:PRINT"Un'altra PARTITA :";
00225 PRINT"(s/n):":INPUT R$
00230 IFR$="n"THENSOUND 4:SOUND 4:END
00240 GOTO 50
01000 INIT 7
01010 PRINT"RIPETI tutta la SERIE di ";
01020 PRINT"NUMERI apparsi"
01030 PRINT"da Inizio GIOCO"
01040 PRINT"Ogni VOLTA ce n'e' uno di";
01050 PRINT" PIU' ?!?":PRINT
01060 PRINT"Batti-RET-per Cominciare";
01070 INPUT R$
01080 RETURN
```

```
RIPETI tutta la SERIE di NUMERI apparsi
da Inizio GIOCO
Ogni VOLTA ce n'e' uno di PIU' ?!?

Batti-RET-per Cominciare
```

Parole

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: gioco di lettere (per 1 o vari giocatori).
Difficoltà del gioco: media.

IL GIOCO:

È difficile immaginare questo gioco più semplice nel suo principio, che si accontenta di visualizzare nel disordine le lettere prese da una trentina di parole. Ma cercate di giocare, e vedrete che non è così facile: la difficoltà varia evidentemente secondo la lista di parole che avrete introdotto in DATA prima.

IL PROGRAMMA:

La linea 40 permette di dimensionare la variabile A$ (I) che conterrà le parole.
Il ciclo I 50 prende la lista delle parole (scritte in DATA alle linee 240÷330).
La linea 70 sceglie una parola a caso; il ciclo I (100-155) controlla se una lettera non è già stata pescata, e riprende le lettere nel disordine.
La linea 160 visualizza le lettere, e le linee 170 e 175 aspettano la vostra proposta, che è testata alla linea 180.
Il punteggio è incrementato alla linea 190.
La fine della partita è controllata dalla linea 210, ed il punteggio è visualizzato alle linee 220 e 225.

PER GIOCARE:

Si lancia il programma con RUN. Una serie di lettere appare allora in disordine, e bisogna cercare di ritrovare la parola corrispondente. Si batte la parola supposta, poi RETURN, ed una nuova serie di lettere apparirà, se si risponde S alla domanda UN'ALTRA PROVA (s/n). Il punteggio ed il numero di tentativi appaiono alla fine della partita.

ESTENSIONI POSSIBILI:

Si può pensare di limitare il tempo di apparizione della parola, o fare sfilare le lettere le une dopo le altre sullo schermo. Si può anche aumentare la difficoltà inserendo dei JOLLY, giocare in molti, tenere il conto dei punti, ecc.

```
00010 PRINT"PAROLE...PAROLE 056@&"
00020 INPUT "Dammi un Num.INTERO :":F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 DIM A$(30):S=0:E=0
00050 FOR I=1 TO 30:READA$(I):NEXT I
00060 INIT 7
00070 Z=INT(RND(1)*30+1)
00080 IFA$(Z)=""GOTO 70
00090 R$=A$(Z):A$="":A$(Z)="":V$=R$
00100 FOR I=1 TO LEN(R$)
00110 Z=INT(RND(1)*LEN(R$)+1)
00120 IFMID$(R$,Z,1)="-"GOTO 110
00130 A$=A$+MID$(R$,Z,1)
00140 Q$=LEFT$(R$,Z-1)+"-"
00150 R$=Q$+RIGHT$(R$,LEN(R$)-Z)
00155 NEXT I:PRINT"Batti:  LOCK"
00160 PRINT:PRINT"LE LETTERE sono :":A$
00165 SOUND 4:SOUND 0:SOUND 2:SOUND 0
00170 CURSORX 1:CURSORY 6
00175 INPUT "La tua SOLUZIONE :":S$
00180 IFS$=V$THENS=S+1:PRINT:GOTO 340
00190 E=E+1
00200 CURSORX 1:CURSORY 12
00205 INPUT"Un'altra PROVA (S/N):":R$
00210 IFLEFT$(R$,1)="S"GOTO 60
00220 CURSORX 1:CURSORY 20:PRINT"Tuoi":
00225 PRINT" PUNTI:";S;" su ";E;" PROVE
00230 SOUND 4:SOUND 0:SOUND 2:END
00240 DATA ERUDITO,ENERGUMEN0,AFFILIATO
00250 DATA SALAME,INSIGNIFICANTE,IDEALE
00260 DATA IMPRESSIONANTE,TREGUA,EPICO
00270 DATA INTREPIDO,CODARDO,ALLETTANTE
00280 DATA CONSACRATO,SENATO,BANDERUOLA
00290 DATA ELABORATORE,TREGUA,CALAMITA
00300 DATA ARBITRARIO,CONSACRATO,DIEDRO
00310 DATA ATTUALE,OPPORTUNO,ALTRUISTA
00320 DATA TERMINE,AUTOGRAFO,NOVIZIO
00330 DATA SPAZZACAMINO,PELURIA,CALORE
00340 PRINT"GIUSTO !?!":RESTORE:PRINT
00350 SOUND 2:SOUND 5:SOUND 5:GOTO 50
```

```
Batti:  LOCK

LE LETTERE sono :LFITIFAOA

La tua SOLUZIONE :? affiliato



Un'altra PROVA (S/N):? s



Tuoi PUNTI: 1 su  1 PROVE
```

Permutazioni

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessione.
Difficoltà del gioco: media.

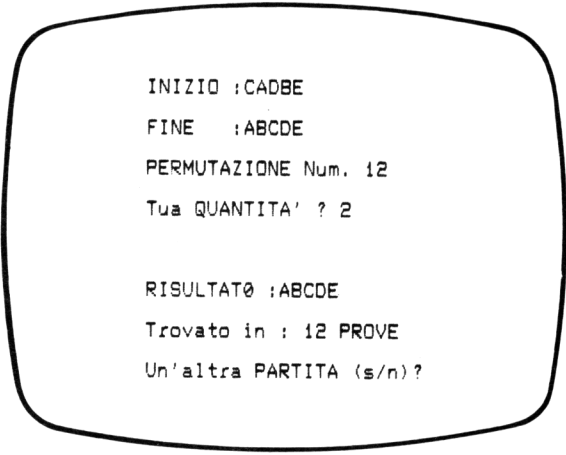
IL GIOCO:

Il Videopac vi chiede prima il numero di lettere che desiderate permutare (non più di 12). Da lì vi visualizzerà il numero corrispondente di lettere prese all'inizio dell'alfabeto, prima nel disordine (INIZIO) poi nell'ordine (FINE). Dovrete riuscire a rimettere, nell'ordine, le lettere date all'inizio, utilizzando delle permutazioni sulle N prime lettere partendo da sinistra, col minor numero di tentativi possibile.

IL PROGRAMMA:

La linea 40 permette di scegliere il numero di lettere, e la linea 50 controlla che non se ne chiedano più di 12.
La linea 80 realizza la mescola, prendendo a caso la nuova posizione di ogni lettera. Le lettere sono in seguito aggiunte man mano che sono pescate, in una variabile B$ (linea 100), eliminate in A$ e rimpiazzate in C$, nell'ordine giusto (linee 110-120).
La linea 140 controlla il numero di lettere di B$.
Le visualizzazioni sono effettuate dalle linee (155÷180).
Il rigiro è fatto, lettera per lettera, nel ciclo I (210-230); il risultato è piazzato in B$, alla linea 240; visualizzato dalle linee 260 e 270 e paragonato al risultato finale alla linea 280.
Il numero di tentativi sarà visualizzato alla fine della partita, grazie alla linea 300.

PER GIOCARE:

Dopo aver battuto RUN, appare la domanda NUMERO DI LETTERE (DA 1 A 12); si risponde battendo N, seguita da RETURN. La visualizzazione seguente presenta l'ordine iniziale (INIZIO) e l'ordine da ottenere (FINE).
Si gioca indicando il numero di lettere da permutare partendo da sinistra, per esempio: su un messaggio iniziale di GEABDFC, se si batte 3 e RETURN, si otterrà AEGBDFC. Si continuerà fino ad ottenere l'ordine desiderato.
Il punteggio apparirà alla fine della partita; così come la domanda UN'ALTRA PARTITA? (s/n); rispondere s se si desidera continuare.

ESTENSIONI POSSIBILI: Si potrà utilizzare lo stesso tipo di gioco per, ad esempio, ricostituire un disegno nascosto, dove le diverse parti sarebbero identificate con delle lettere. Si possono anche realizzare delle permutazioni più complesse (per esempio, il fatto di permutare due lettere permuta anche le due vicine alla lettera più a sinistra, ecc.)

```
00010 PRINT"- PERMUTAZIONI - 0570&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXTI:PRINT
00040 INPUT "Numero LETTERE (1 a 12)";N
00050 IF N>12 THEN40
00060 A$="ABCDEFGHILMN"
00065 C$="            ";B$=""
00070 INIT 7:PAGE
00080 X=INT(RND(1)*N+1)
00090 M$=MID$(A$,X,1):IF M$=" "THEN 80
00100 B$=M$+B$
00110 G$=LEFT$(C$,X-1)+M$
00115 C$=G$+RIGHT$(C$,12-X)
00120 E$=LEFT$(A$,X-1)+" "
00125 A$=E$+RIGHT$(A$,12-X)
00130 S=1
00140 IFLEN(B$)<>NTHEN80
00150 C$=LEFT$(C$,LEN(B$)):CURSORY 4
00160 PRINT"INIZIO :";:TX1,4,0:PRINTB$
00165 TX4,0,0:CURSORY 6
00170 PRINT"FINE   :";:TX4,4,0:PRINTC$
00175 TX4,0,0:CURSORY 8:
00180 PRINT"PERMUTAZIONE Num.";S:PRINT
00185 INPUT "Tua QUANTITA' ";L
00190 IF L>N THEN175
00195 S=S+1
00200 A$=""
00210 FOR I=1 TO L
00220 A$=A$+MID$(B$,L-I+1,1)
00230 NEXT I
00240 B$=A$+RIGHT$(B$,LEN(B$)-L)
00250 SOUND 2:CURSORY 14:PRINT"RISUL";
00255 PRINT"TAT0 :";:TX5,4,0:PRINTB$
00260 TX4,0,0:IFB$<>C$THEN175
00270 SOUND 2:SOUND 4:PRINT
00275 PRINT"Trovato in :";S-1;" PROVE"
00280 PRINT:PRINT "Un'altra PARTITA ";
00285 INPUT "(s/n)";R$
00290 IFLEFT$(R$,1)="s"THENPRINT:GOTO40
```

```
INIZIO :CADBE

FINE   :ABCDE

PERMUTAZIONE Num. 12

Tua QUANTITA' ? 2


RISULTAT0 :ABCDE

Trovato in : 12 PROVE

Un'altra PARTITA (s/n)?
```

Bridge

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: variabile, secondo il problema posto.

IL GIOCO:

Non troverete, in questo programma, un compagno per i vostri bridge, ma, più esattamente, una simulazione per la visualizzazione di una mano. Questa visualizzazione è realizzata gioco per gioco; si può, partendo da questo, porsi ogni sorta di problemi: annunci, gioco della carta, ecc.

IL PROGRAMMA:

Le linee 50 e 60 definiscono un mazzo di 52 carte.
La variabile B$, alla linea 70, permetterà di dare in abbreviato i nomi dei colori e anche di migliorare la nomina di certe carte (10, fante, donna, re e asso).
Il ciclo J (80÷240) realizza la pesca di quattro mazzi di 13 carte.
Il ciclo I (90÷190) realizza pesca e stampa. La carta è scelta tra le 52 disponibili, alla linea 100.
La linea 110 calcola il colore della carta pescata.
La linea 120 verifica che questa carta non sia già stata pescata.
Le linee 150 e 170 si occupano della stampa complementare nel caso si peschi un 10, un fante, una donna, un re o un asso.
La stampa è realizzata dalle linee 155, 160 e 175.
Le linee 180 e 185 permettono di eliminare le carte già pescate, e le linee 200 e 210 realizzano un arresto tra ogni visualizzazione.

PER GIOCARE:

Si lancia il programma con RUN. Una prima serie di 13 carte appare verticalmente. Basterà battere il tasto RET per ottenere la seconda serie di 13 carte. Si procederà nello stesso modo per la terza e la quarta serie.

ESTENSIONI POSSIBILI:

Si può immaginare di raffigurare le 13 carte in una maniera più convenzionale. Si può anche aggiungere la possibilità di fare degli annunci (e di paragonarli a quelli del Videopac, per esempio), o introdurre delle possibilità di gioco della carta, compreso anche il Videopac.

```
00010 PRINT"BRIDGE 0058@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:CLEAR250
00050 R$="234567891FDRA"
00060 R$=R$+R$+R$+R$
00070 B$="FiQuCuPi0ADES"
00080 FOR J=1 TO 4
00090 FOR I=1 TO 13
00100 X=INT(RND(1)*LEN(R$)+1)
00110 S=2*INT((X-1)/13)+1
00120 IFMID$(R$,X,1)="0"GOTO 100
00130 T=X-13*INT(X/13)
00140 T=T-13*(T=0)
00150 CURSORX 10*J-9:CURSORY I
00155 PRINTMID$(R$,X,1);
00160 IFT>=9THENPRINTMID$(B$,T,1)
00170 CURSORX 10*J-6:CURSORY I
00175 PRINTMID$(B$,S,2)
00180 Q$=LEFT$(R$,X-1)+"0"
00185 R$=Q$+RIGHT$(R$,LEN(R$)-X)
00190 NEXT I
00200 CURSORX 1:CURSORY 18
00210 INPUT "Con-RET-continui";S$
00220 CURSORX 1:CURSORY 18:PRINT"      ";
00230 PRINT"                   "
00240 NEXT J
```

```
RE Pi    FA Qu    7  Pi    RE Cu
8  Pi    10 Cu    2  Cu    9  Fi
10 Qu    8  Cu    AS Pi    DO Cu
3  Qu    5  Pi    5  Qu    2  Fi
8  Fi    5  Cu    9  Qu    3  Cu
RE Qu    4  Qu    6  Pi    AS Fi
9  Cu    FA Pi    6  Qu    FA Fi
FA Cu    6  Cu    9  Pi    AS Cu
RE Fi    4  Fi    5  Fi    DO Fi
4  Pi    4  Cu    3  Fi    AS Qu
DO Qu    7  Fi    DO Pi    2  Pi
8  Qu    10 Pi    7  Qu    6  Fi
3  Pi    2  Qu    7  Cu    10 Fi
```

Hanoi

RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: difficile.

IL GIOCO: È presentato, qui, in una versione per un giocatore. Si dispone di un certo numero di dischi (8 al massimo) e si deve trasferirli dalla torre numero 1 alla torre numero 3, rispettando una sola regola: in nessun momento un disco può essere ricoperto da un disco di diametro superiore. Il numero di tentativi non è contabilizzato in questo programma ma, invece, le mosse illecite saranno rifiutate.

IL PROGRAMMA: La variabile D$, alla linea 30, servirà a disegnare i dischi. Il numero di dischi è inserito alla linea 50 e rifiutato se ci sono più di 6 dischi. Il ciclo I (70-90) permette di fabbricare una variabile A(I) che contiene lo stato della pila sotto forma numerica (alla linea 30, tre variabili sono previste per conservare lo stato delle tre pile).
Le basi sono disegnate dalle linee (105÷115) e le tre pile dai cicli chiusi I (130÷180) e J (140÷170).
Il controllo della mossa è effettuato dalla linea 270 e lo stato delle pile è modificato dalle linee 275 e 280.

PER GIOCARE: Dopo un RUN che lancia il programma, si inserisce il numero di dischi richiesti, seguito da RETURN. Si ottiene il disegno dello stato iniziale e, alla domanda DISCO DI PARTENZA, si risponde dando il numero della torre di partenza. Questo numero (1, 2 o 3) sarà rifiutato se la torre è vuota. Poi si batte RETURN e, alla domanda DISCO D'ARRIVO, si risponde col numero della torre d'arrivo. Se la mossa non è possibile, saremo obbligati di ricominciare da DISCO DI PARTENZA. Se la mossa è possibile, lo spostamento è realizzato e si ricomincia.

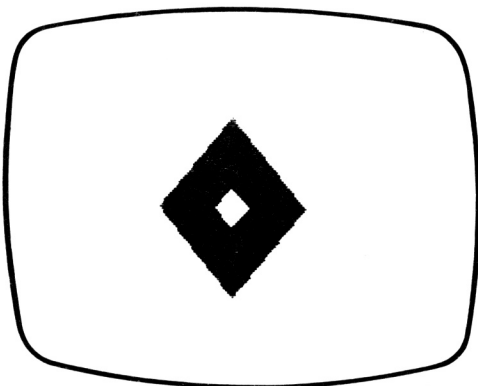
ESTENSIONI POSSIBILI: Si potranno disegnare dei dischi di diversi colori, introdurre un contatore di mosse e, soprattutto, definire un gioco per il calcolatore. Si può anche cercare di complicare il gioco con delle torri in più e due giocatori, ognuno avente dei dischi di colore diverso, per esempio.

```
00010 PRINT" -* H A N O I *-  0590&"
00020 INIT 7:CLEAR500:D$=""
00030 DIM AZ(3):AZ(1)=9:AZ(2)=9:AZ(3)=9
00040 FOR I=1 TO 9:D$=D$+CHR$(72):NEXTI
00050 INPUT"Num.DISCHI (Meno di 6):";N
00060 IF N>=6 THEN50
00070 FOR I=0 TO N-1
00080 AZ(1)=VAL(STR$(AZ(1))+STR$(N-I))
00090 NEXT I
00100 INIT 7:PAGE:STORE
00105 CURSORY13:TX5,4,0:PRINTD$
00110 CURSORX13:CURSORY13:PRINTD$
00115 CURSORX25:CURSORY13:PRINTD$
00120 CURSORX 5:CURSORY14:TX0,0,0
00125 PRINT"1";SPC(11);"2";SPC(11);"3"
00130 FOR I=0 TO N-1
00140 FOR J=1 TO 3
00150 B=LEN(STR$(AZ(J)))-1
00155 B$=RIGHT$(STR$(AZ(J)),B)
00160 IFLEN(B$)>I+1THENTX0,4,0:GOSUB300
00170 NEXT J
00180 NEXT I:SCREEN:DISPLAY
00190 TX4,0,0:CURSORY 16
00200 INPUT "DISC0 di Partenza:";D
00210 IF D>3 OR D<1 THEN 190
00220 IF AZ(D)=9 THEN 190
00230 PRINT:INPUT"DISCO d'Arrivo   :";A
00240 B=LEN(STR$(AZ(D)))-1
00245 B$=RIGHT$(STR$(AZ(D)),B)
00250 C=LEN(STR$(AZ(A)))-1
00255 C$=RIGHT$(STR$(AZ(A)),C)
00260 M$=MID$(B$,LEN(B$),1)
00265 N$=MID$(C$,LEN(C$),1)
00270 IF M$>N$ THEN 190
00275 AZ(A)=VAL(C$+MID$(B$,LEN(B$),1))
00280 AZ(D)=VAL(LEFT$(B$,LEN(B$)-1))
00290 SOUND 4:GOTO 100
00300 CURSORX 12*J-9:CURSORY 12-I
00305 T=VAL(MID$(B$,I+2,1))
00310 PRINTLEFT$(D$,T):RETURN
```

```
   H
   HH
   HHH
   HHHH
   HHHHH
HHHHHHHHH   HHHHHHHHH   HHHHHHHHH
    1           2           3

DISC0 di Partenza:? 1

DISCO d'Arrivo   :? 2
```

Farfalle

RIFERIMENTI:	Difficoltà di programmazione: facile. Categoria del gioco: intermezzo/passatempo. Difficoltà del gioco: nessuna.
IL GIOCO:	Il programma costruisce progressivamente sullo schermo una farfalla stilizzata e poi analogamente la cancella e così via finché non vi stufate. Potete pensare d'utilizzarlo per addormentarvi, se il gioco della cavallina non vi basta più. La creazione dell'immagine è del tutto casuale.
IL PROGRAMMA:	Un primo ciclo K (50-130) definisce le dimensioni dei successivi tracciati. Il secondo ciclo I (60-120) esegue il tracciato stampando, in senso antiorario, il carattere "127" ossia un rettangolino nero. Appena terminato il disegno inizia la cancellazione, mediante il carattere "132", anch'essa casuale, e poi si riinizia da capo, senza fine ma sempre in modo diverso.
PER GIOCARE:	Si dà inizio all'esecuzione del programma tramite RUN, poi basta stare a guardare. Per fermare il programma occorre battere RESTORE.
ESTENSIONI POSSIBILI:	Si potrà introdurre il suono e/o il colore, e utilizzare l'istruzione LINE per accelerare la formazione o la dissoluzione dell'immagine. Si potrà girare la farfalla di 90°, costruirla a bande invece che ad evolvente, invertire il senso di formazione dell'immagine. Si potranno introdurre dei parametri per verificare le possibilità del Videopac, ed immaginare su principi analoghi tracciati diversi (caleidoscopio, tappezzerie, ecc...).

```
00010 PRINT"FARFALLE 00600&"
00020 INPUT "Dammi un Num.INTERO :":F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:S=0
00050 FOR K=1 TO 10
00060 FOR I=0TO K:Z=INT(RND(1)*5+1)
00070 IFS=0GOTO 80
00075 ONZGOTO 160,170,180,190,200
00080 CURSORX 19+K-1:CURSORY 11+I
00085 PRINTCHR$(127);
00090 CURSORX 19+K-1:CURSORY 11-I
00095 PRINTCHR$(127);
00100 CURSORX 19-K+1:CURSORY 11-I
00105 PRINTCHR$(127);
00110 CURSORX 19-K+1:CURSORY 11+I
00115 PRINTCHR$(127);
00120 NEXT I
00130 NEXT K
00140 S=NOTS
00150 GOTO 50
00160 CURSORX 19-K+1:CURSORY 11-I
00165 PRINTCHR$(32);
00170 CURSORX 19-K+1:CURSORY 11+I
00175 PRINTCHR$(32);
00180 CURSORX 19+K-1:CURSORY 11+I
00185 PRINTCHR$(32);
00190 CURSORX 19+K-1:CURSORY 11-I
00195 PRINTCHR$(32);
00200 GOTO 120
```

Matteo

RIFERIMENTI: Difficoltà di programmazione: facile.
Categoria del gioco: disegno aleatorio.
Difficoltà del gioco: nessuna.

IL GIOCO: Si utilizza la definizione di una curva in coordinate polari sullo schermo dei disegni aleatori, del tipo fiore, più o meno eleganti.

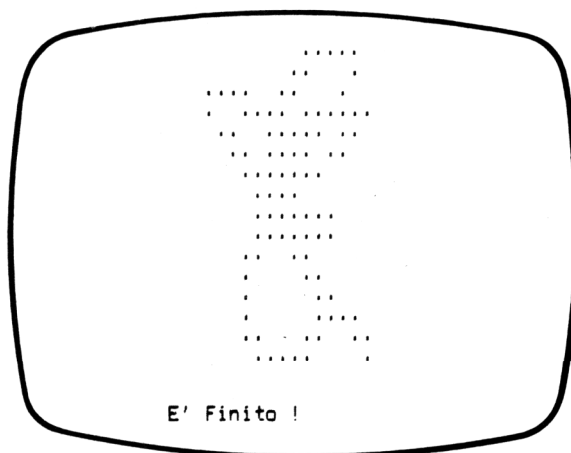
IL PROGRAMMA: La linea 30 permette di scegliere l'origine della memoria di visualizzazione, e la linea 40 fissa il più alto indirizzo disponibile per il Basic.
La linea 50 riempie la memoria colore HR con degli 8.
La linea 60 fa una cancellazione della memoria di visualizzazione HR (che va da 8192 a 16191, ossia 8000 bytes).
La linea 65 mette il Videopac in modo HR.
Il ciclo M (80-140) definisce la curva in coordinate polari, e le linee 100 e 110 garantiscono la conversione in coordinate cartesiane.
Le linee 120 e 130 chiamano i due sottoprogrammi di calcolo della posizione di un punto.
La linea 150 controlla la fine della partita.

PER GIOCARE: Si lancia il programma con RUN e si aspetta pazientemente che l'inizializzazione sia terminata. Il disegno è allora effettuato automaticamente. Si potranno modificare i colori cambiando i numeri contenuti nelle linee 50 e 60. Per fare un altro disegno, basta battere RUN ed attendere.

ESTENSIONI POSSIBILI: L'accesso all'alta risoluzione apre la strada a delle immense possibilità: sarà prudente all'inizio cominciare l'esplorazione di questo campo facendo tracciare al Videopac delle curve matematiche più o meno complesse, ispirandosi al metodo dato qui.

Nota: Poiché l'alta risoluzione non è possibile sul Videopac, il disegno è poco variato e l'andamento non è curvilineo ma a spezzate.

```
00010 PRINT"MATTEO 061@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 Z=2:K=4:T=3:INIT 7
00050 C=INT(RND(1)*8):F=7-C:SCREEN
00060 A=3*RND(1):B=3*RND(1)
00070 FOR H=0 TO 6.28STEP3.14/180
00080 R=Z*SIN(H*T)
00090 X=10+INT(K*R*COS(A*H))
00100 Y=10+INT(K*R*SIN(B*H))
00110 CURSORX X:CURSORY Y
00120 PRINTCHR$(46):NEXT H
00130 CURSORX 1:CURSORY 20:PRINT"E' ";
00140 PRINT"Finito !":SOUND 2:SOUND 5
00150 IFKEY(0)=0 GOTO 40
00160 C=INT(RND(1)*8:IFC=0GOTO 170
00170 GOTO 60
```

E' Finito !

Il lupo e gli agnelli

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: media.

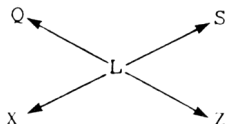
IL GIOCO:

È un gioco abbastanza conosciuto che può essere praticato su una scacchiera 8*8, o, indifferentemente, su una 10*10. È presentato, qui, in una versione per una scacchiera 8*8; uno dei giocatori sceglie il lupo che si sposta seguendo le diagonali, restando sulle caselle nere, di una casella ad ogni colpo, e che può avanzare o retrocedere. Gli agnelli si spostano ugualmente sulle caselle nere, in diagonale, ma non hanno il diritto di retrocedere. Le caselle già occupate sono proibite (niente prese), e non ci sono salti.
Il lupo vince se riesce ad arrivare sulla linea in basso prima che tutti gli agnelli occupino quella in alto; gli agnelli vincono in questo caso, o se riescono a bloccare il lupo.

IL PROGRAMMA:

Una variabile A$, alla linea 30, contiene le posizioni di partenza. I cicli I (80÷130) e J (90÷120) disegnano la scacchiera e le linee da 135 a 205 posizionano il lupo e gli agnelli. La linea 210 gestisce il cambio di giocatore. La linea 320 visualizza "TURNO DEL LUPO" se è il giocatore che manipola il lupo che deve giocare, e la linea 330 "TURNO DEGLI AGNELLI" nel caso contrario. Le linee da 245 a 260 analizzano la risposta, in funzione del giocatore in questione.

PER GIOCARE:

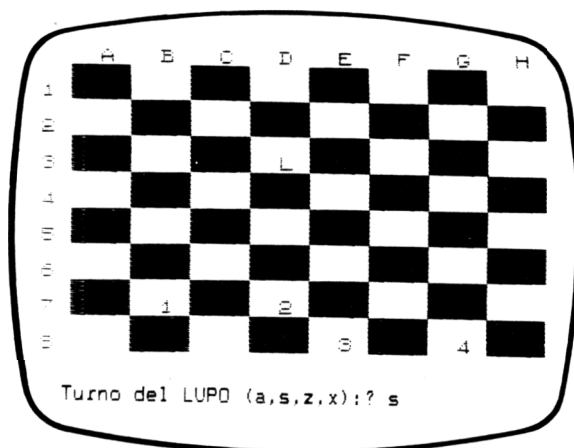
RUN lancia l'arresto sull'attesa di un ingresso: 0 e RETURN se si vuole che gli agnelli comincino, e 1 per il lupo. Dopodiché, la scacchiera e la frase TURNO DEL LUPO (A,S,Z,X) per il lupo e TURNO DEGLI AGNELLI (ND E NS) per gli agnelli appaiono. Gli agnelli rispondono con un numero, seguito da D o S (per destra o sinistra). Così 2D provoca lo spostamento dell'agnello numero 2 che sale in diagonale di una casella a destra. Lo spostamento effettuato è visualizzato, poi si ottiene una domanda identica per il lupo che risponderà battendo Q, S, X o Z, secondo la direzione scelta:

Si dovrà fare attenzione a non uscire dalla scacchiera e a non finire sull'avversario, nessun controllo essendo previsto.

ESTENSIONI POSSIBILI: Si potrà prevedere una individuazione delle caselle, un grafismo, più curato, dei controlli di non superamento dei limiti del tavolo, di non arrivo su una casella già occupata...
Si potrà anche pensare di giocare su una scacchiera di 10*10 caselle, che aumenta leggermente la difficoltà del gioco per gli agnelli. Ma l'estensione più interessante da realizzare sarà certamente la creazione di un programma che permetta al Videopac di comportarsi come un avversario onorevole, sia col lupo, sia con gli agnelli (e, in questo caso, sarà possibile definire una strategia vincente per il Videopac).

```
00010 PRINT"LUPO e AGNELLI  0620&"
00020 INPUT "AGNELLI (0) o LUPO (1)";S
00030 A$="8183858714"
00040 INIT 7:PAGE:STORE
00050 CURSORY1:PRINT"    A   B   C  ";
00055 PRINT" D   E   F   G   H"
00060 FOR I=1 TO 8:CURSORY 2*I+1
00065 PRINTI:NEXT I
00070 Z$=CHR$(127)
00080 FOR I=1 TO 8
00090 FOR J=1 TO 8
00100 CURSORX 4*J:CURSORY 2*I
00105 IF(I+J)/2<>INT((I+J)/2)THEN120
00110 PRINTZ$+Z$+Z$+Z$:CURSORX 4*J
00115 PRINTZ$+Z$+Z$+Z$
00120 NEXT J
00130 NEXT I:SCREEN:DISPLAY
00135 CURSORX 4*VAL(MID$(A$,10,1))-2
00140 CURSORY 2*VAL(MID$(A$, 9,1))+1
00145 PRINT"L"
00150 CURSORX 4*VAL(MID$(A$, 2,1))+5
00155 CURSORY 2*VAL(MID$(A$, 1,1))+1
00160 PRINT"1"
00165 CURSORX 4*VAL(MID$(A$, 4,1))+5
00170 CURSORY 2*VAL(MID$(A$, 3,1))+1
00175 PRINT"2"
00180 CURSORX 4*VAL(MID$(A$, 6,1))+5
00185 CURSORY 2*VAL(MID$(A$, 5,1))+1
00190 PRINT"3"
00195 CURSORX 4*VAL(MID$(A$, 8,1))+5
00200 CURSORY 2*VAL(MID$(A$, 7,1))+1
00205 PRINT"4"
00210 S=ABS(S-1)
00215 CURSORY20:PRINT"                    ";
00220 PRINT"                    "
00225 CURSORY20:IFS=0THENGOSUB 320
00230 IFS=1THEN330
00240 C=10
00245 A=VAL(RIGHT$(A$,2))
00250 A=A+(C+1ANDR$="x")+(C-1ANDR$="z")
```

(*segue*)

(*seguito*)

```
00260 A=A-(C+1ANDR$="a")-(C-1ANDR$="s")
00270 GOTO 300
00280 C=2*VAL(MID$(R$,1,1))
00285 A=VAL(MID$(A$,C-1,2))
00290 A=A-(9ANDRIGHT$(R$,1)="d")
00295 A=A-(11ANDRIGHT$(R$,1)="s")
00300 E$=LEFT$(A$,C-2)
00305 O$=E$+RIGHT$(STR$(A),2)
00310 A$=O$+RIGHT$(A$,LEN(A$)-C)
00315 GOTO 80
00320 PRINT"Turno del LUPO (a,s,z,x):";
00325 INPUT R$:STORE:RETURN
00330 PRINT"Turno degli AGNELLI (Nd/";
00335 INPUT "Ns) :";R$:STORE:GOTO 280
```

LIVELLO 4

N.	Nome del programma	Tema e particolarità
63	Lancio di dadi	Abbastanza flessibile da poter esser utilizzato ovunque.
64	Tombola	Vincete senza problema alla prossima estrazione.
65	La bomba	Sapreste scoprirla in tempo?
66	Lavagna magica	Sbizzarritevi nelle vostre creazioni grafiche, senza carta né MATITA.
67	Pubblicità	Se proprio non sapete che cosa esporre in vetrina.
68	Evasione	Dovrete avere un bel po' di fortuna per riuscirvi.
69	Cappelli	Chi è buon osservatore riuscirà.
70	Duello aereo	Non è così facile abbattere un aereo che si muove in continuazione.
71	Disegno di dadi	Migliorate i vostri giochi di dadi inserendovi questo programma.
72	Il gioco del missile	Una missione assai pericolosa per un solo giocatore.
73	La scimmia	Gioco di lettere per due giocatori, in cui servono dei buoni riflessi.
74	Il gioco del 15	Gioco numerico di riflessione, per due giocatori.
75	Destrezza	Gioco di riflessi visivi, per un giocatore.
76	Sottomarino	Un classico videogioco per un solo giocatore.
77	Zombi	Gioco visivo un po' bizzarro, per un solo giocatore.
78	Diagonale	Avete proprio dei buoni riflessi?
79	Decomaco	Per evitarvi di star in secca troppo a lungo con Nicomaco.
80	Caccia al sottomarino	Vi occorre non poca riflessione e molta fortuna.
81	Tastiera	Imparate a battere a macchina divertendovi.
82	La lucertola d'Alice	Assai spettacolare, ma con molto rischio.
83	Calcolatrice	Programma di simulazione molto realistico.

Animazione

PRESA AL VOLO DI UNO O PIÙ CARATTERI BATTUTI ALLA TASTIERA: KEY(0)

Il principio d'un gioco grafico d'animazione è che esso si deve svolgere in un certo senso, da solo, per un certo periodo di tempo, finché non intervenga il giocatore.
Bisogna quindi disporre della possibilità d'osservare se c'è stato o no l'intervento del giocatore, senza però interrompere lo svolgimento del programma.
Nel Basic del VIDEOPAC+ questa possibilità è offerta dalla funzione: KEY(0), che esegue lo scrutamento dei tasti e registra il codice di carattere dell'ultimo tasto premuto oppure riporta 0 se nessun tasto viene battuto.
Per ottenere il carattere alfanumerico si farà per esempio A$ = CHR$(KEY(0)) che registra il nome dell'ultimo tasto abbassato; si potrà d'altra parte registrare in A$ una sequenza di caratteri in ordine di battuta (è necessario che ci sia la battuta: non si può lasciare il dito premuto sul tasto in questione).
Si potrà utilizzare CHR$(KEY(0)) in due modi diversi:

— sia in un ciclo (realizzazione d'un movimento). Si cattura KEY(0) e lo si confronta con l'eventuale uscita dal ciclo;
— sia in un ciclo d'attesa: finché A$ è vuoto (non è stato battuto niente) s'aspetta, e s'agisce non appena un tasto sia stato battuto.

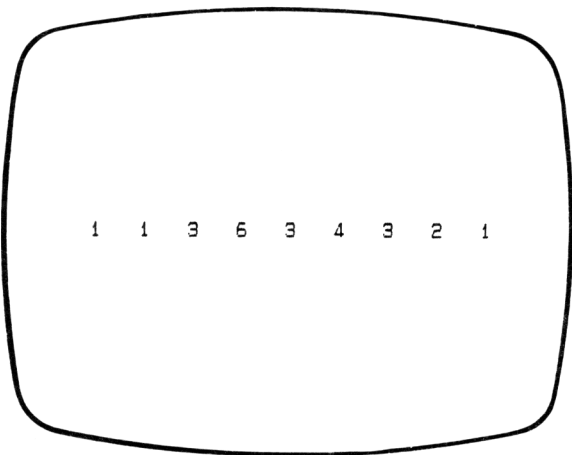
Talvolta può essere necessario battere il tasto più volte affinché esso venga preso in considerazione. Per facilitare la lettura del tasto, in certi programmi sarà vantaggiosa una breve temporizzazione di ritardo.

Nei programmi di giochi, salvo indicazione contraria, i tasti presi per gli spostamenti nelle quattro direzioni sono:

(si batte in realtà ./.)	per gli spostamenti	a sinistra
=		a destra
+		verso l'alto
—		verso il basso.

Si potranno facilmente modificare i programmi, se si desiderano utilizzare altri tasti.

SPOSTAMENTO DI OGGETTI DAL BASSO VERSO L'ALTO DELLO SCHERMO: SIMULAZIONE DELLO SCROLL

Il VIDEOPAC+ realizza automaticamente uno SCROLL verso l'alto quando si scrive sull'ultima riga in basso dello schermo, questa proprietà è stata usufruita in numerosi giochi in quanto consente di spostare di una linea verso l'alto vari oggetti (testi o grafici) presenti ad un certo istante sullo schermo; la riga più alta viene evidentemente perduta. Se un oggetto deve rimanere immobile sullo schermo, occorrerà cancellarlo dalla sua nuova ubicazione rimpiazzarlo nella posizione iniziale.
È comunque difficile lavorando in Basic, gestire lo spostamento simultaneo di più oggetti sullo schermo (altrimenti si ottiene presto un'animazione a scatti, a causa di un tempo di spostamento proibitivo).
La specifica inversa dello SCROLL è PAGE che ha la proprietà di fissare lo schermo. È stata usata talvolta quando per necessità di gioco venivano coinvolte le linee proibite dello schermo (0, 21 e 22).

Lancio di dadi

RIFERIMENTI: Difficoltà di programmazione: nessuna.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: nessuna.

IL GIOCO: Il programma proposto non realizza un gioco vero e proprio, ma potrà essere utilizzato come sottoprogramma per numerosi giochi. Offre semplicemente la possibilità di lanciare da 1 a 9 dadi, con la semplice pressione di un tasto. I dadi sono in seguito visualizzati con un grafismo molto semplificato.

IL PROGRAMMA: La linea 60 prende il carattere battuto sulla tastiera e lo trasforma in numero (se non è possibile, il programma rifiuta l'ingresso).
Una variabile D non dimensionata riceve il valore indicato da ogni dado.
Un ciclo J (110÷130) è utilizzato per disegnare un fondo grigio, ed un altro ciclo I (100-160) serve al lancio (lancio alla linea 140 e stampa alla linea 160).
Le linee 160 e 170 offrono la possibilità di rifare un lancio; non è indispensabile.

PER GIOCARE: Si lancia il programma con l'ordine RUN e si ottiene il messaggio NUMERO DI DADI DA LANCIARE? al quale si risponde battendo una cifra. Il risultato è allora automaticamente visualizzato; basta allora battere un tasto qualunque per ricominciare il programma. Per fermarsi, è necessario battere RESTORE.

ESTENSIONI POSSIBILI: Si potrà, naturalmente, utilizzare questo programma per la realizzazione di ogni gioco utilizzante dei dadi (421, ecc.)

```
00010 PRINT"Tiro di DADI 063@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 CURSORY3:PRINT"Num.DADI da Tirare
00060 A$=CHR$(KEY(0)):IFA$="" THEN60
00070 IF A$<"1" OR A$>"9" THEN40
00080 N=VAL(A$)
00090 INIT 7
00100 FOR I=1 TO N
00110 FOR J=1 TO 3:TX3,0,0
00120 CURSORX4*I:CURSORY8+J:PRINT"  "
00130 NEXT J
00140 D(I)=INT(RND(1)*6+1)
00150 CURSORX 4*I:CURSORY 10:TX1,0,0
00160 PRINTRIGHT$(STR$(D(I)),1):NEXT I
00170 TX4,0,0:SOUND 2:GOTO 50
```

```
1  1  3  6  3  4  3  2  1
```

Tombola

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: fortuna (vari giocatori).
Difficoltà del gioco: facile.

IL GIOCO:

Si tratta di realizzare una partita completa di tombola, cioè di ottenere 6 numeri, compresi tra 1 e 49, più un numero complementare, tutti differenti, che potrebbero essere, per esempio, utilizzati per riempire la vostra prossima cartella di tombola.

IL PROGRAMMA:

Si utilizzerà una tabella N (7), dimensionata alla linea 40, per registrare i risultati.
La pesca dei numeri è effettuata in un ciclo I (60÷120); un secondo ciclo J (90÷110) è utilizzato per controllare che non ci siano due pescate identiche.
La pescata (o la ripescata) è effettuata alla linea 70.
Il ciclo I (170÷220) visualizza il risultato, e la linea 240 visualizza il numero complementare.
Le linee da 230 a 250 si occupano del caso in cui si vorrebbe rifare una partita.

PER GIOCARE:

Basta battere RUN per lanciare il programma e ottenere la visualizzazione dei primi numeri.
La domanda VUOI UN ALTRO TIRO? (s/n) appare nel basso dello schermo; si risponderà s se si desidera continuare.

ESTENSIONI POSSIBILI:

Si può immaginare un programma che potrebbe tenere conto sia dei propri risultati, sia dei risultati inseriti da un giocatore, per migliorare la pescata, tenendo conto delle probabilità di avere un dato numero.
Si può anche immaginare un programma che realizzerebbe dei cartoncini per la tombola più usuale, come quella che è ancora correntemente praticata di solito.

```
00010 PRINT"T O M B O L A  0640&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 DIM N(7)
00050 INIT 7
00060 FOR I=1 TO 7
00070 N(I)=INT(RND(1)*59+1)
00080 IFI=1GOTO 120
00090 FOR J=1 TO I-1
00100 IFN(I)=N(J)THENJ=I-1:NEXTJ:GOTO70
00110 NEXT J
00120 NEXT I
00130 CURSORX 5:CURSORY 3:TX0.2.0
00140 PRINT"T O M B O L A - L O T T O "
00150 CURSORX 5:CURSORY 4
00160 PRINT"---------------------------"
00170 FOR I=1 TO 6
00180 CURSORX 6*I-5:CURSORY 10:TX0.3.0
00190 PRINTN(I);:PRINTN(I)
00200 CURSORX 6*I-5:CURSORY 11
00210 PRINTN(I);:PRINTN(I)
00220 NEXT I:
00230 CURSORX 5:CURSORY 17:TX0.1.0
00240 PRINT"Numero COMPLEMENTARE:";N(7)
00250 CURSORX 5:CURSORY 18
00260 PRINT"Numero COMPLEMENTARE:";N(7)
00270 SOUND 5:SOUND 4:SOUND 2:SOUND 0
00280 CURSORX 1:CURSORY 20:TX0.0.0
00290 PRINT"Vuoi un altro TIRO (s/n):":
00300 INPUT R$
00310 IFR$="s"GOTO 50:GOTO 40
00320 TX0.4.0:PRINT"F I N E":END
```

La bomba

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO:

Dovete trovare, con l'aiuto di un detector speciale, un pacco radioattivo deposto da un viaggiatore, male intenzionato, in una camera d'albergo. Un solo problema, di misure: l'albergo possiede un milione di camere (100 * 100 * 100).
Il vostro detector visualizza, in permanenza, un segnale che vi informa sulla prossimità della bomba, e il tempo che vi resta, decontabilizzato di 10 in 10.
Potrete scoprire la bomba prima dell'esplosione?

IL PROGRAMMA:

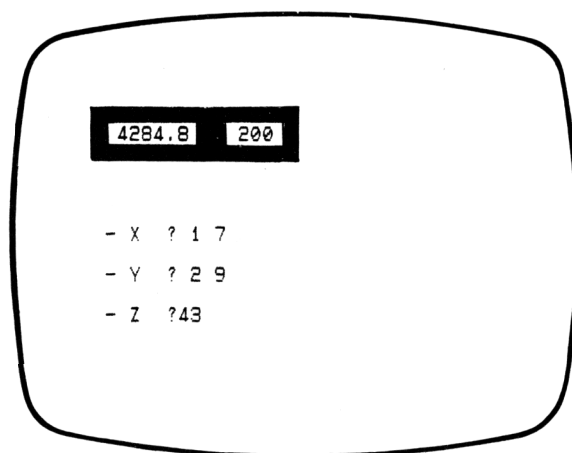
Una variabile numerica R, non dimensionata, contiene le tre coordinate che definiscono la posizione del giocatore.
La posizione della camera contenente il pacco è definita da una pesca aleatoria di 3 variabili A, B, C, alla linea 50.
Il ciclo I (70-110) disegna il detettore.
Il segnale di prossimità è calcolato alla linea 130 e la linea 140 permette la visualizzazione delle coordinate della camera da scoprire, nel caso che si abbia perso.
Le linee 150 e 155 permettono la visualizzazione sia del segnale che del tempo rimanente.
La linea 165 realizza il conto alla rovescia del tempo.
La linea 170 controlla se si è riusciti a trovare o no.
Il ciclo I (210÷250) permette di inserire le nuove coordinate, utilizzando il sottoprogramma posizionato alle linee da 300 a 310.

PER GIOCARE:

Dopo il lancio del programma con RUN, il segnale ed il tempo (all'inizio 200) appaiono. Il giocatore è allora alla posizione X=0, Y=0, Z=0, e si dovranno introdurre successivamente le tre coordinate X,Y,Z della camera da esplorare.
Si farà, per esempio, 13, poi RETURN, 45, poi RETURN, ecc.
Il nuovo segnale è allora visualizzato, ed il tempo decrementato di 10. Si continuerà così sia fino al buon risultato (arresto del gioco e fine del programma), sia fino all'apparizione di 3 numeri (X,Y,Z) senza uscire dal programma, il che significa allora che abbiamo perso.

ESTENSIONI POSSIBILI: Sarà senz'altro indispensabile migliorare il grafismo, aggiungere un'esplosione (sonora e visiva) per simbolizzare l'esplosione della bomba, visualizzare simbolicamente la posizione nell'albergo, ecc.

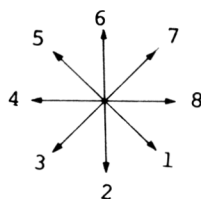
```
00010 PRINT"LA BOMBA 065@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 L=100:T=200
00045 A=INT(RND(1)*L)
00050 B=INT(RND(1)*L)
00055 C=INT(RND(1)*L)
00060 INIT 7:PAGE:STORE
00070 FOR I=1 TO 18
00080 CURSORXI:CURSORY4:PRINTCHR$(127)
00090 CURSORXI:CURSORY5:PRINTCHR$(127)
00100 CURSORXI:CURSORY6:PRINTCHR$(127)
00110 NEXT I:SCREEN:DISPLAY
00120 S=A/L+B+C*L-(R(1)/L+R(2)+R(3)*L)
00125 S=ABS(S)
00130 S$=LEFT$(STR$(S),7):S=VAL(S$)
00140 IFT=0THENPRINT"PERSO . "GOTO 270
00150 CURSORX 2:CURSORY 5:PRINTS
00155 CURSORX12:CURSORY 5:PRINTT
00160 SOUND6:SOUND2:SOUND3
00165 T=T-INT(SQR(L))
00170 IFS=0THENCURSORX 1:GOTO 280
00180 CURSORX1:CURSORY10:PRINT" - X  :"
00190 CURSORX1:CURSORY12:PRINT" - Y  :"
00200 CURSORX1:CURSORY14:PRINT" - Z  :"
00210 FOR I=1 TO 3
00220 R$=""
00230 GOSUB 300
00240 R(I)=VAL(R$)
00250 NEXT I
00260 GOTO 60
00270 PRINT"La Soluzione :";A;"  ";B;
00275 PRINT"  ";C:SOUND2:SOUND3:GOTO150
00280 CURSORY 20:PRINT"BRAVO !!"
00290 SOUND6:SOUND4:SOUND3:SOUND2:END
00300 CURSORX 7:CURSORY 2*I+8:PRINT"?";
00310 PRINTR$;"    ":SOUND 4
00320 A$=CHR$(KEY(0)):IFKEY(0)=0THEN320
00330 IFKEY(0)=13THENRETURN
00340 R$=R$+A$
00350 GOTO 300
```

```
 4284.8   200

- X  ? 1 7

- Y  ? 2 9

- Z  ?43
```

Lavagna magica

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: disegno.
Difficoltà del gioco: facile.

IL GIOCO:

Come indica il titolo, si tratta, semplicemente, di utilizzare lo schermo come una lavagna, manipolando un punto sullo schermo seguendo 8 direzioni, con l'aiuto dei tasti da 1 a 8, secondo la convenzione:

Si dispone anche della possibilità di spostare il punto sullo schermo senza lasciare la traccia battendo 0 (il che permette la cancellazione parziale) e lasciare di nuovo la traccia ribattendo 0. Si può scegliere il colore del punto battendo su un tasto da A a M; è possibile cancellare tutto lo schermo battendo P.

IL PROGRAMMA:

La linea 30 contiene la scelta delle possibilità di colore da 1 a 7.
Le linee 50 e 55 stampano il punto elementare.
Le linee da 60 a 90 controllano gli ingressi.
La linea 130 chiama uno dei 4 sottoprogrammi che effettuano lo spostamento.
La linea 140 verifica che non si esca dai limiti dello schermo.
La linea 160 effettua la cancellazione se richiesta.
Il sottoprogramma 400÷410 sceglie il colore; 300÷310 realizza o no la cancellazione.

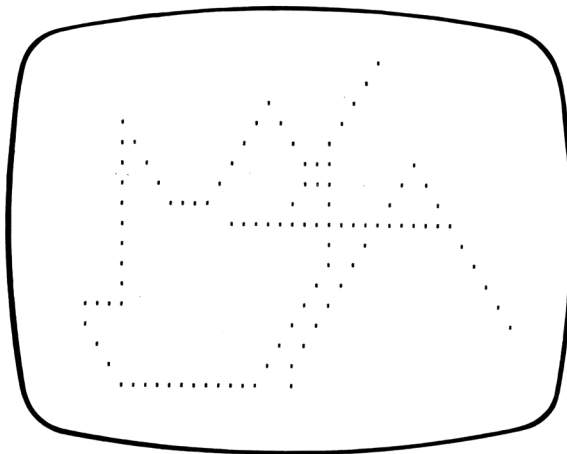
PER GIOCARE:

Si lancia il programma con RUN; un punto appare verso la metà dello schermo. Si dovrà battere 0 per spostarlo senza visualizzare la traccia. Se si batte una lettera da A a M, il colore del punto cambia. Il tasto P cancella tutto, e si ricomincia. Si può uscire in ogni momento, battendo RESTORE.

ESTENSIONI POSSIBILI:

Si potrà, naturalmente, pensare di scrivere un programma identico utilizzando l'alta risoluzione, prevedere un sottoprogramma che per-

metta la salvaguardia del disegno realizzato per un'ulteriore utilizzazione, ecc.

```
00010 PRINT"LAVAGNA MAGICA 066@&"
00020 INIT 7:PAGE
00030 C$="01234567"
00040 X=20:Y=12:C=4
00050 CURSORX X:CURSORY Y:TXC,0,0
00055 PRINTCHR$(46)
00060 A$=CHR$(KEY(0)):IFA$=""GOTO60
00070 IFA$>="A"ANDA$<="H"THENGOSUB 400
00080 IFA$="0"THENGOSUB 300
00090 IFA$="P"THEN20
00100 F=-2*(VAL(A$)<5)-1
00110 A=X:B=Y
00120 U=VAL(A$)+2*(F-1)
00130 ONUGOSUB 1000,2000,3000,4000
00140 D=-(X>0ANDX<40ANDY>1ANDY<22)
00150 IFD=0THENGOSUB 5000
00160 IFS=0GOTO 50
00170 CURSORXA:CURSORYB:PRINTCHR$(46)
00180 GOTO 50
00300 S=ABS(S-1)
00310 RETURN
00400 C=VAL(MID$(C$,ASC(A$)-48,1))
00410 RETURN
01000 Y=Y+F
02000 X=X+F
02010 RETURN
03000 GOSUB 2000
04000 Y=Y-F
04010 RETURN
05000 X=A:Y=B
05010 RETURN
```

Pubblicità

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: visualizzazione pubblicitaria.
Difficoltà del gioco: nessuna.

IL GIOCO:

Un testo qualunque dovrà essere introdotto all'inizio del programma. Potrà essere relativamente lungo (4 o 5 linee); si sceglieranno in seguito il senso e la velocità di scorrimento: al seguito di ciò il testo apparirà (una sola linea per volta) e comincerà a scorrere alla velocità scelta e nel senso scelto.

IL PROGRAMMA:

Il testo sarà introdotto lettera per lettera alle linee (60÷100).
Il messaggio sarà limitato a 36 caratteri grazie alle linee 200 e 210.
Lo scorrimento verso sinistra è preparato alla linea 150, e lo scorrimento verso destra alla linea 160.
Il testo sarà visualizzato alla linea 140, e la durata della visualizzazione (velocità di scorrimento) è scelta dalla linea 170.
La linea 190 permetterà di cambiare, se ce ne sarà bisogno, le condizioni dello scorrimento.

PER GIOCARE:

Si lancerà il programma con RUN e, all'apparizione della domanda INSERITE IL VOSTRO TESTO, si introdurrà il testo da fare scorrere, seguito da RETURN. Si ottiene in seguito il messaggio SCORRIMENTO A DESTRA (D) O A SINISTRA (S) al quale si risponde battendo sia D, sia S seguiti da RETURN.
Poi, alla domanda VELOCITÀ, si risponderà introducendo un piccolo numero per una velocità grande, e un grande numero per una velocità piccola (per esempio, da 0 a 40 o più).
Il testo comincerà a sfilare appena dopo il RETURN. Se si desiderano cambiare le condizioni dello scorrimento, basta battere il tasto 0 per ritrovare le domande SENSO e VELOCITÀ. Si batterà RESTORE per interrompere il programma.

ESTENSIONI POSSIBILI:

Si potrà introdurre il colore, realizzare una visualizzazione ondulante, fare risalire le linee sullo schermo, ecc.

```
00010 PRINT"Affissione/RECLAME  067@&"
00020 INIT 7:CLEAR 500:PAGE
00030 PRINT"BATTI il Tuo TESTO"
00050 PRINT:PRINT
00060 A$=CHR$(KEY(0))
00070 FOR I=1 TO 99:NEXT I
00080 IF A$=CHR$(0) THEN 60
00090 IF ASC(A$)=13 THEN 120
00100 B$=B$+A$+A$
00110 TX0,2,0:PRINTA$;" ";:GOTO 60
00120 TX0,0,0:PRINT
00130 PRINT:PRINT"Scorrimento: DESTRA";
00140 PRINT" o SINISTRA (d/s)";:INPUTS$
00150 PRINT:INPUT"Velocita'(0 a 10):";V
00160 V=11-V
00170 INIT 7
00180 T=LEN(B$)
00190 CURSORY 10:TX1,3,0:STORE
00200 PRINT" "+MID$(B$,1,36)+" ";" "
00210 PRINT" "+MID$(B$,1,36)+" ";" "
00220 IF LEFT$(S$,1)<>"s" THEN 240
00230 B$=RIGHT$(B$,T-1)+LEFT$(B$,1)
00240 IF LEFT$(S$,1)<>"d" THEN 260
00250 B$=RIGHT$(B$,1)+LEFT$(B$,T-1)
00260 FOR J=1 TO V:SCREEN:DISPLAY
00270 C$=CHR$(KEY(0))
00280 IF C$="0" THEN J=V:NEXT J:GOTO120
00290 NEXT J
00300 GOTO 190
```

Evasione

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: solitario.
Difficoltà del gioco: facile.

IL GIOCO:

Il vostro solo problema: riuscire ad attraversare lo schermo, senza che il Videopac vi abbia prima colpito con uno dei suoi numerosi esplosivi che vi tirerà addosso. Potete spostarvi in 4 direzioni (con l'aiuto delle 4 frecce), e dovete riuscire ad uscire dal terreno dal lato destro, senza, naturalmente, essere mai stati colpiti.

IL PROGRAMMA:

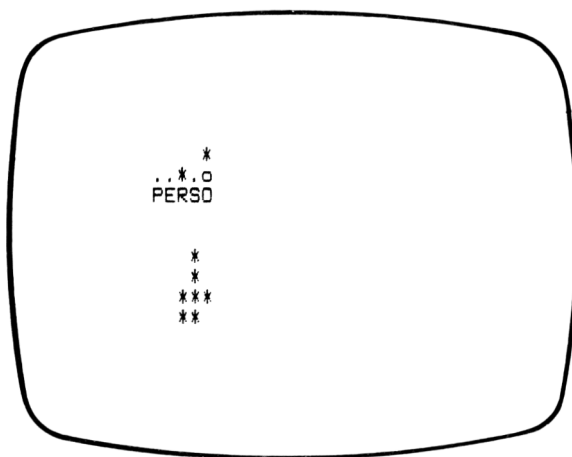
La linea 50 definisce la posizione di partenza del giocatore.
La linea 60 sceglie la coordinata orizzontale della casella dove il Videopac sparerà (legata alla vostra posizione), e la linea 70 sceglie la coordinata verticale (arbitraria) di questa casella.

PER GIOCARE:

Si comincia l'esecuzione del programma con RUN. Un primo sparo ha luogo, dopo che il giocatore sia apparso, raffigurato da una picca. I 4 tasti di direzione (>,<, freccia in su, freccia in giù) servono a spostare il giocatore. Se è colpito, si ottiene, sullo schermo, il messaggio PERSO e la partita si ferma. Se riesce ad uscire alla destra dello schermo senza essere colpito, si ottiene il messaggio BRAVO, e la partita è ugualmente terminata. Si può terminare in ogni momento battendo RESTORE.

ESTENSIONI POSSIBILI:

Il colore ed un grafismo più curato miglioreranno considerabilmente l'attrattiva del gioco. Si potrà anche rendere il lancio delle bombe più aleatorio, introdurre degli ostacoli da aggirare, aggiungere dei controlli, fare partecipare vari giocatori, ecc.

```
00010 PRINT"- E V A S I O N E - 068@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 X=1:Y=11
00060 H=(X-1)+3*RND(1)
00070 K=20*RND(1)+1
00080 IFX<3THEN120
00090 FOR J=1 TO 7*RND(1)+3
00100 CURSORX H+3*RND(1)-2
00105 CURSORY K+3*RND(1)-2
00110 PRINT"*":NEXT J
00120 TX4,0,0:SOUND 4
00125 CURSORX X:CURSORY Y:PRINT"."
00130 A$=CHR$(KEY(0))
00140 X=X-(A$=CHR$(61))+(A$=CHR$(47))
00150 Y=Y-(A$=CHR$(45))+(A$=CHR$(43))
00160 CURSORX X:CURSORY Y:PRINT"o"
00170 IFX>3THEN200
00180 IFX<38THEN60
00190 PRINT"BRAVO":SOUND 2:END
00200 Z=ABS(H-X)+ABS(K-Y)
00210 IFZ>3THEN180
00220 PRINT"PERSO":SOUND7:END
```

Cappelli

RIFERIMENTI:

Difficoltà di programmazione: abbastanza facile.
Categoria del gioco: gioco di riflessi, per due giocatori.
Difficoltà del gioco: facile.

IL GIOCO:

L'idea è molto semplice: due personaggi si spostano l'uno verso l'altro sullo schermo e si incrociano. Nel momento dove si incrociano, è possibile che scambino i loro cappelli. Si tratta di essere il primo a dire se c'è stato o no lo scambio di cappelli durante l'incrocio.

IL PROGRAMMA:

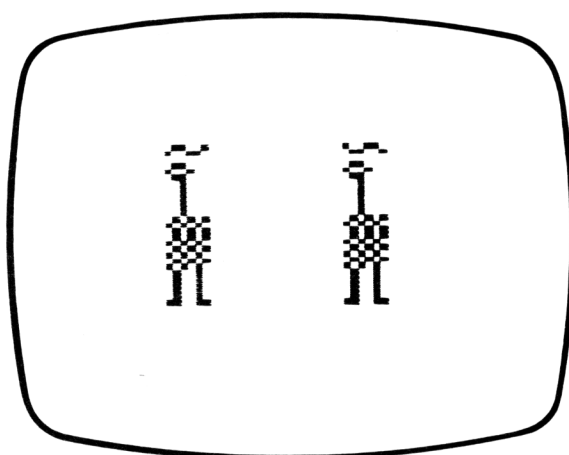
La linea 40 contiene il disegno dei due cappelli (in due variabili alfanumeriche B$ e C$).
La linea 50 contiene, in un'altra variabile A$, i simboli che permetteranno di disegnare i personaggi.
La linea 70 permette, se è il caso, lo scambio di cappelli.
La variabile T della linea 80 serve ad individuare se c'è stato o no lo scambio.
Il disegno dei due personaggi è realizzato da un ciclo I (120÷150), e il loro spostamento è realizzato dalla linea 160.
Il test, che permette o no lo scambio dei cappelli, è effettuato alla linea 180.
Le linee da 200 a 250 determinano se il primo giocatore che ha battuto su un tasto ha perso o vinto.

PER GIOCARE:

Lanciare il programma con RUN. I personaggi cominciano immediatamente a spostarsi l'uno verso l'altro. Il giocatore di sinistra dispone dei tasti 1 e 2 e quello di destra, dei tasti 9 e 0. Se c'è stato cambio di cappello, i tasti da premere saranno 1 (a sinistra) e 0 (a destra). Se non c'è stato cambio, bisognerà premere 2 (a sinistra) e 9 (a destra).
Apparirà allora il messaggio BRAVO Sinistra o BRAVO Destra, o, ancora, PERSO Sinistra o PERSO Destra.

ESTENSIONI POSSIBILI:

Sarà possibile prevedere una visualizzazione a colori, con un'animazione dei personaggi. Si potrà anche complicare il gioco a piacere con vari oggetti scambiati in maniera casuale (bisognerà allora trovarne la lista completa).

```
00010 PRINT"- C A P P E L L I - 069@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 B$="KLG":C$="NCM"
00050 A$="f9  5  5 ffffff999j 5z u"
00060 X=8:Y=30:T=1
00070 D$=C$:C$=B$:B$=D$
00080 T=ABS(T-1)
00090 INIT 7:PAGE:GR0,7,0:STORE
00100 CURSORX X:CURSORY 3:PRINTC$
00110 CURSORX Y:CURSORY 3:PRINTB$
00120 FOR I=1 TO 8
00130 GR1,7,0::CURSORX X:CURSORY I+3
00135 PRINTMID$(A$,3*I-2,3)
00140 GR5,7,0:CURSORX Y:CURSORY I+3
00145 PRINTMID$(A$,3*I-2,3)
00150 NEXT I:SCREEN:DISPLAY
00160 X=X+1:Y=Y-1:IFX>36THENX=36
00165 IFY<1THENY=1
00170 FOR I=1 TO 200:NEXT I
00180 IFX=Y AND RND(1)<.5 THEN 70
00190 R$=CHR$(KEY(0))
00195 IFR$=CHR$(0)THEN90
00200 TX4,0,0:PRINT
00210 IFX<=YTHEN230
00220 IF(R$="0" AND T=1) THEN270
00225 IF(R$="1" AND T=0) THEN270
00230 IF(R$="9" AND T=1) THEN280
00235 IF(R$="8" AND T=0) THEN280
00240 IF R$="1" OR R$="0" THEN290
00250 IF R$="8" OR R$="9" THEN300
00260 SOUND 4:SOUND 0:END
00270 PRINT"BRAVO Sinistra":SOUND 2:END
00280 PRINT"BRAVO Destra":SOUND 2:END
00290 PRINT"PERSO Sinistra":GOTO 260
00300 PRINT"PERSO Destra":GOTO 260
```

Duello aereo

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessi (1 giocatore).
Difficoltà del gioco: difficile.

IL GIOCO:

Al centro dello schermo, si trova il mirino del vostro aereo, e l'aereo nemico si sposta davanti a voi, in una maniera che potrà sembrarvi all'inizio incoerente. Potete spostarvi in tutte le direzioni, grazie alle frecce di movimento, e dovrete cercare di portare l'aereo nemico al centro del mirino. Sparate battendo 0, e vedrete i vostri proiettili che convergono verso di lui. Ma continua a spostarsi anche durante il vostro sparo, e non è per niente evidente che riusciate a colpirlo.
La partita continuerà a svolgersi fino a che l'avrete colpito (messaggio VINTO) o che sia uscito dal vostro campo visivo.

IL PROGRAMMA:

La linea 75 disegna l'aereo nemico il cui spostamento aleatorio sarà realizzato dalle linee 90 e 100.
Le linee (120÷135) vi permettono il pilotaggio del vostro aereo.
Le linee (165÷175) disegnano il mirino del vostro aereo.
Il sottoprogramma, che gestisce lo sparo, va dalla linea 200 alla linea 250, ed il test di fine partita è effettuato alla linea 240.
La linea 180 guarda se avete sparato o no.

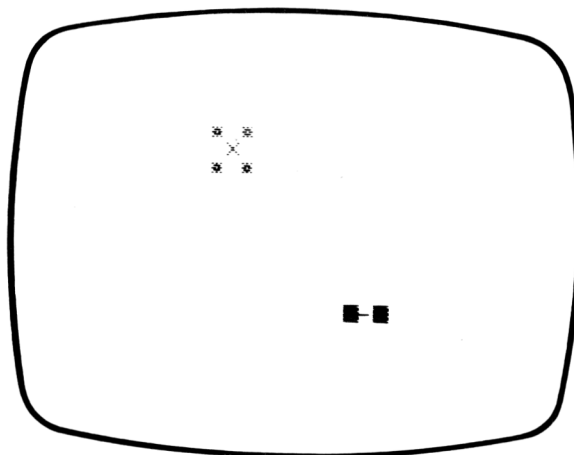
PER GIOCARE:

Appena il programma è stato lanciato con l'ordine RUN, dovete manovrare per cercare di inquadrare l'apparecchio nemico nel vostro mirino. Per questo, disponete delle frecce di spostamento (tasti >, <, freccia su e giù, freccia destra sinistra) e potete sparare in ogni istante premendo il tasto 9.
L'apparecchio nemico può continuare a spostarsi in maniera aleatoria, anche quando avete sparato.
La partita si terminerà con la visualizzazione di VINTO, se colpite l'aereo nemico, o, eccezionalmente, dal fatto che l'aereo nemico sia uscito dai limiti dello schermo. Si può anche fermare in ogni momento battendo RESTORE.

ESTENSIONI POSSIBILI:

Sarà abbastanza facile utilizzare una grafica più curata, utilizzando il colore. Si potrà anche pensare di stabilizzare la posizione dell'ae-

reo nemico durante lo sparo (basta modificare la linea 180). Possiamo anche introdurre vari aerei, contare i punti od il tempo, tenere conto della scorta di carburante, ecc.

```
00010 PRINT"DUELLO AEREO 070@&"
00020 INPUT"Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE
00050 A=5:B=15
00060 N=20
00070 CURSORX B:CURSORY A
00075 PRINT"o"+CHR$(45)+"o"
00080 U=A:V=B
00090 A=A+2*INT(RND(1)*2)-1
00100 B=B+2*INT(RND(1)*2)-1
00110 A$=CHR$(KEY(0))
00120 F=2*(A$=CHR$(61))-2*(A$=CHR$(47))
00125 B=B-F
00130 E=2*(A$=CHR$(45))-2*(A$=CHR$(43))
00135 A=A-E
00140 IFB<1ORB>37THENB=ABS(B-37)
00150 IFA<3ORA>21THENA=ABS(A-21)
00160 CURSORX V:CURSORY U:PRINT"   "
00165 CURSORX 18:CURSORY 9:PRINT"* *"
00170 CURSORX 18:CURSORY 10:PRINT" X "
00175 CURSORX 18:CURSORY 11:PRINT"* *"
00180 IFA$="9"ORN<20THENGR0,7,0:GOTO200
00190 GOTO 70
00200 CURSORX 29-N:CURSORY N:PRINT"Z"
00205 CURSORX N+9:CURSORY N:PRINT"e"
00210 CURSORX 29-N:CURSORY N:PRINT"@"
00215 CURSORX N+9:CURSORY N:PRINT"@"
00220 N=N-1:TX0,0,0
00230 IF N>9 THEN 70
00240 IF(A=9ORA=10ORA=11)THEN 260
00250 GOTO 60
00260 IF(B=18ORB=19ORB=20)THEN280
00270 GOTO 60
00280 PRINT"VINTO !!":SOUND4:SOUND0:END
```

Disegno di dadi

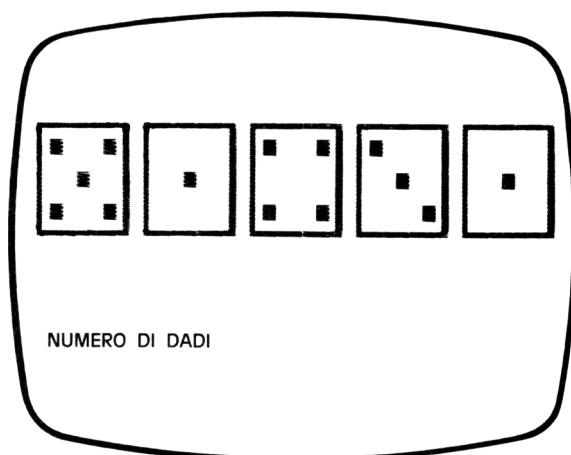
RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: lancio di dadi.
Difficoltà del gioco: facile.

IL GIOCO: Il programma proposto è soprattutto interessante per il suo lato grafico visto che permette di raffigurare, in maniera più che discreta, da 1 a 4 dadi sullo schermo, utilizzando una tecnica di disegno un po' particolare.
Ogni risposta alla domanda NUMERO DADI? provoca immediatamente un nuovo lancio del numero indicato di dadi.

IL PROGRAMMA: La linea 60 permette di controllare se c'è stato o no intervento su un tasto da 1 a 4.
Le linee 90 e 100 inizializzano le due variabili di disegno dei dadi.
Il ciclo I (110÷210) e le linee (120÷135) disegnano la cornice del dado.
La linea 140 realizza il lancio, ed il ciclo J (150÷200) disegna i punti in relazione al lancio realizzato (chiamando le variabili A$ e B$).

PER GIOCARE: Si lancia il programma con RUN, e, alla domanda NUMERO DI DADI, si risponde battendo un tasto da 1 a 4. I dadi richiesti sono allora raffigurati, e la domanda NUMERO DADI è di nuovo posta. Basterà battere RESTORE per uscire dal programma.

ESTENSIONI POSSIBILI: Si potrà conservare la parte del disegno utilizzata in questo programma per realizzare tutta una serie di giochi con i dadi, con una rappresentazione grafica di qualità discreta e facile da modificare se ce ne fosse bisogno.

```
00010 PRINT"DISEGNO di DADI 071@6"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:CLEAR2000
00050 CURSORX1:CURSORY15
00055 PRINT"Numero DADI ?"
00060 A$=CHR$(KEY(O)):IFA$=""GOTO 60
00070 N=VAL(A$):IFN<1ORN>5GOTO 50
00080 INIT 7:C$=CHR$(127)
00085 D$="       "+C$+"  "+C$
00090 A$=D$+"        "+C$+C$+"   "+C$
00100 B$="121314324515525555"
00110 FOR I=1 TO N:GR0,7,0
00120 CURSORX 8*I-7:CURSORY 2
00125 PRINT"WCCCCCk"
00130 CURSORX 8*I-7:CURSORY 8
00135 PRINT"upppppz":TX0,0,0
00140 X=INT(RND(1)*6+1)
00150 FOR J=1 TO 5
00160 IFJ<=3THENGOSUB 220
00170 GR0,7,0
00180 CURSORX8*I-7:CURSORYJ+2:PRINT"U"
00190 CURSORX8*I-1:CURSORYJ+2:PRINT"j"
00200 TX0,0,0:NEXT J
00210 NEXT I:GOTO 50
00220 S=5*VAL(MID$(B$,3*(X-1)+J,1))-4
00230 CURSORX 8*I-6:CURSORY 2*J+1
00240 PRINTMID$(A$,S,5):RETURN
```

Il gioco del missile

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: contro il Videopac.
Difficoltà del gioco: media.

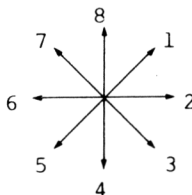
IL GIOCO:

Il Videopac è in allarme: siete ai comandi di un aereo incaricato di una missione di spionaggio sul suo territorio; il vostro scopo è di sorvolare la base nemica (raffigurata da una H reverse), e di ritornare sani e salvi alla vostra base; ma, appena il Videopac vi avrà avvistati, manderà, al vostro inseguimento, un missile molto perfezionato, che non avrà pace fino a che non vi avrà distrutto. Il vostro aereo può spostarsi in tutte le direzioni, grazie ai tasti da 1 a 8; riuscirete a portare a termine la vostra missione?

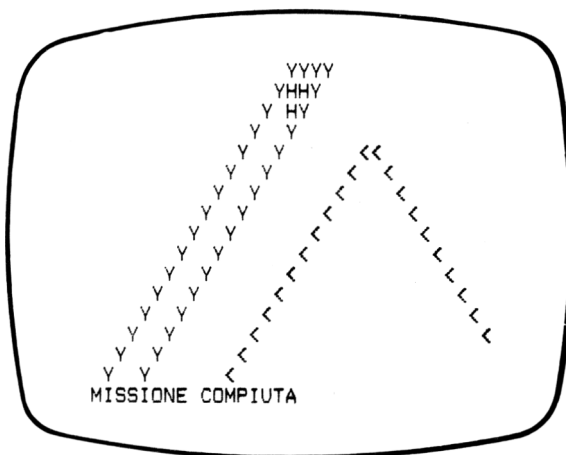
IL PROGRAMMA:

Le linee 50 e 55 disegnano la base nemica da sorvolare (sotto forma di una H in negativo).
Le linee 60 e 65 disegnano la vostra base di partenza. Le linee 80 e 100 scelgono l'istante di partenza del missile.
La linea 120 disegna il vostro aereo.
Le linee 140÷155 definiscono il suo movimento seguendo 8 direzioni, utilizzando i sottoprogrammi alle linee 1000, 2000, 3000 e 4000.
La linea 160 assicura la partenza del missile che è disegnato dalla linea 170 ed il cui spostamento è gestito dalle linee 180÷190. Le linee 200÷210 controllano la fine della partita.

PER GIOCARE:

Lanciare il programma con RUN.
Il campo di gioco appare e potete spostare il vostro aereo utilizzando i tasti da 1 a 8, con la convenzione seguente:

Dovete farlo girare attorno al territorio nemico (se voi, o il missile, toccate questo territorio, una seconda partita comincia). Se il missile vi distrugge, ottenete un messaggio COLPITO, e la partita si ferma. Battere RESTORE per fermare la partita.

ESTENSIONI POSSIBILI: Sarà senz'altro necessario aggiungere vari controlli nel programma (l'aereo ha sorvolato la base nemica, è tornato alla propria base?...) Si possono anche prevedere vari livelli di gioco facendo, per esempio, variare la velocità del missile, prevedere l'utilizzo di trucchi per l'aereo, l'esplosione del missile dopo un certo tempo, ecc.

```
00010 PRINT"Gioco del MISSILE 072@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:M=0:TX0,3,0:Z$=CHR$(127)
00050 CURSORX 17:CURSORY 5:PRINT"HH"
00055 CURSORX 17:CURSORY 6:PRINT"HH"
00060 CURSORX 1:CURSORY 20:PRINTZ$+" ";
00065 PRINTZ$+" ":PRINTZ$+" "+Z$+" "
00070 TX0,0,0:U=15:X=18:PAGE
00080 S=INT(RND(1)*U+6)
00090 Y=1
00100 S=S-1
00110 V=33
00120 CURSORX Y:CURSORY X:PRINT"Y"
00130 A$=CHR$(KEY(0)):IFA$=""GOTO 130
00140 F=-2*(VAL(A$)<5)-1
00150 Z=VAL(A$)+2*(F-1)
00155 ONZGOSUB 1000,2000,3000,4000
00160 IFS<>0GOTO 100
00170 CURSORX V:CURSORY U:PRINT"<"
00180 U=U-SGN(U-X)
00190 V=V-SGN(V-Y)
00200 IFU=XANDV=YTHENPRINT"COLPITO...":END
00210 IF(U=4ORU=5)AND(V=17ORV=18)GOTO40
00220 IFX<=4ANDY>=18THENM=1
00230 IFM=1ANDX=18ANDY<=4GOTO 250
00240 GOTO 120
00250 PRINT"MISSIONE COMPIUTA":SOUND 4:END
00260 SOUND 2:SOUND 0:SOUND 4
01000 X=X-F
02000 Y=Y+F
02010 RETURN
03000 GOSUB 2000
04000 X=X+F
04010 RETURN
65520 REM-Screen COPY on G7420TURBO/vcg
65521 PAGE:SOUND 2:CURSORY0:PRINT".";
65522 PRINTTAB(39);".":CURSORY 22
65523 PRINT".";TAB(39);".":S=1
65524 CURSORY 1-S
65525 FOR I=0 TO 21+S
65526 FOR J=1 TO 39
65527 C=PEEK(2*J+80*I-32760)
65528 LPRINTTAB(J);CHR$(C);
65529 NEXT J:LPRINT:NEXT I:RETURN
```

La scimmia

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: due giocatori.
Difficoltà del gioco: difficile.

IL GIOCO:

Si ispira all'idea già anziana della scimmia che, battendo a caso su una macchina per scrivere, ha delle possibilità non nulle di finire per scrivere un testo sensato.
Il ruolo della scimmia è, qui, preso dal Videopac che farà apparire dei gruppi di 3 a 6 lettere durante un tempo relativamente corto. Se uno dei giocatori stima che può riuscire a fare una parola con le lettere proposte, deve battere rapidamente sul suo tasto di gioco (1 o 0).
La parola dovrà allora essere accettata o rifiutata dall'avversario che risponderà GIUSTO o FALSO. Ogni parola giusta vale 1 punto, ogni parola sbagliata leva 1 punto.

IL PROGRAMMA:

La linea 40 contiene una lista V$ delle vocali correnti, e la linea 45 una lista C$ delle consonanti correnti.
Sarà possibile modificare queste liste per aumentare, per esempio, la frequenza di apparizione di certe lettere (potremo così mettere due E in V$, o due R in C$, ecc.)
La linea 50 definisce la lunghezza della parola (in principio, da 3 a 6 lettere, ma si può ugualmente modificare questa lunghezza).
Il ciclo I (80÷130) permette la creazione di una "parola" prendendo una lettera, sia nell'alfabeto completo, sia nella lista C$ (linea 110), sia nella lista V$ (linea 115).
La linea 150 temporizza l'apparizione del risultato.
La linea 170 gestisce la fine della partita, con visualizzazione del punteggio.
La linea 180 guarda se uno dei due giocatori ha reagito.
Infine, le linee (200÷215) contabilizzano i punti di ogni giocatore.

PER GIOCARE:

Si lancia il programma con RUN. La prima serie di lettere è allora visualizzata sullo schermo. Il giocatore di sinistra dovrà battere 1 se vede che può realizzare una parola, quello di destra dovrà battere 0.
Se non è stato battuto niente, un'altra serie di lettere sarà visualizzata dopo un po' di tempo.
Se uno dei giocatori ha battuto 0 o 1, la do-

manda (G)IUSTO o (F)ALSO appare allora. Si risponde G se la parola è corretta, S se è sbagliata, seguiti da RETURN. I punti sono allora calcolati dal Videopac, e la serie di lettere seguente è allora visualizzata. Per uscire dal programma, si dovrà battere P al posto di 0 o 1: il punteggio finale è allora visualizzato.

ESTENSIONI POSSIBILI: Si potrà modificare la lunghezza della parola, la frequenza di apparizione delle lettere, la decontabilizzazione dei punti, la regola del gioco, ecc.

```
00010 PRINT"La SCIMMIA 0730&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 A=0:B=0:V$="AEIOU"
00045 C$="BCDGLMNPRSTV"
00050 X=INT(RND(1)*4+3)
00060 M$=""
00070 INIT 7
00080 FOR I=1 TO X
00090 E$=CHR$(INT(RND(1)*26+65))
00100 O=INT(LEN(C$)*RND(1)+1)
00105 U=INT(LEN(V$)*RND(1)+1)
00110 IFRND(1)<.4THENE$=MID$(C$,O,1)
00115 IFRND(1)<.4THENE$=MID$(V$,U,1)
00120 M$=M$+E$
00130 NEXT I
00140 CURSORX 1:CURSORY 6:TX0,3,0
00145 PRINTM$+" ";PRINTM$+" "
00150 FOR J=1 TO 500:NEXT J
00160 TX0,0,0:A$=CHR$(KEY(O))
00170 IFA$="9"THENCURSORX1:GOTO 230
00180 IFA$<>"0"ANDA$<>"1"GOTO 50
00190 PRINT:INPUT"(G)iusto/(F)also:";R$
00200 C=A-(R$="G"ANDA$="0")
00205 A=C+(R$="F"ANDA$="0")
00210 D=B-(R$="G"ANDA$="1")
00215 B=D+(R$="F"ANDA$="1")
00220 GOTO 50
00230 CURSORY10:PRINT"PUNTEGGIO di A:";
00240 PRINTA;" ; PUNTEGGIO di B :";B
00250 SOUND 4:SOUND 0:SOUND 2:END
```

```
ALBE

(G)iusto/(F)also:? G
```

Il gioco del 15

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione (2 giocatori).
Difficoltà del gioco: media.

IL GIOCO:

È un gioco per due giocatori, abbastanza vecchio, che era un tempo praticato nelle fiere. Si dispone di un tappeto di gioco di 9 caselle, e ognuno scommette a turno un gettone del suo colore, su una casella disponibile. Il primo dei due giocatori che, con esattamente 3 gettoni, riesce a realizzare un totale di 15 ha vinto.

IL PROGRAMMA:

Il ciclo I (50÷80) disegna il tavolo di gioco, e le linee 95÷105 si occupano della gestione del turno dei giocatori (numero 1 o 2 per ogni giocatore, e gettone O o X corrispondente).
La linea 110 attende il colpo giocato.
Il ciclo I (120÷140) verifica la validità del colpo.
La linea 150 testa la fine della partita, in caso di partita nulla, e la linea 160 rifiuta i colpi non regolamentari (con ritorno alla situazione iniziale, linea 90).
Le linee 170 e 180 si occupano della visualizzazione del colpo, e le linee 190 e 195 lo registrano in una variabile R$. La parte più interessante del programma, che potrebbe servire per definire una strategia per la macchina, va dalla linea 210 alla linea 300: si utilizzano 3 cicli chiusi per verificare se uno dei giocatori ha ottenuto, con 3 gettoni un totale di 15. Il cambio del giocatore si effettua alla linea 350.

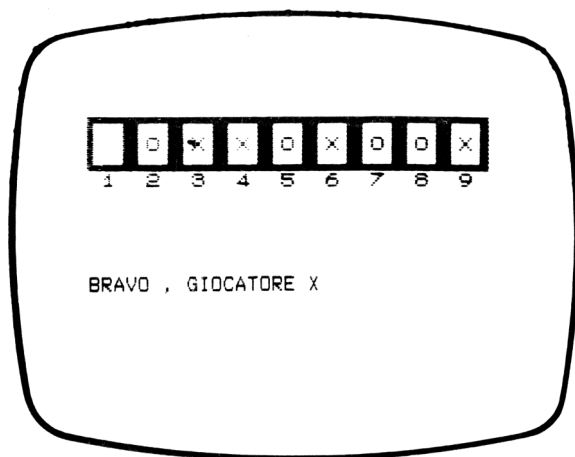
PER GIOCARE:

Lanciare il programma battendo RUN. Il campo di gioco appare, poi la domanda GIOCATORE 1: X, e il giocatore, che ha scelto le X, dovrà dare il numero della casella dove desidera giocare; se è libera, il suo colpo vi sarà registrato. Si ottiene, in seguito, il messaggio GIOCATORE 2: O ed il secondo giocatore procede nella stessa maniera. Se uno dei giocatori vince (totale di 15 in tre colpi), ottiene il messaggio BRAVO e la partita si ferma. In caso di partita nulla, la macchina visualizza È FINITA. Si farà un'altra partita battendo RUN.

ESTENSIONI POSSIBILI:

La più interessante sarà senz'altro di cercare di definire una strategia di macchina (cercan-

do di evitare che la macchina giochi sempre il 5 quando deve cominciare). Si può anche, naturalmente, migliorare il grafismo ed introdurre degli effetti sonori per i differenti messaggi.

```
00010 PRINT"Il GIOCO del 15 074@&"
00020 INIT 7
00030 F=0:R$="123456789"
00040 O$="":X$="":PAGE:STORE
00050 FOR I=0 TO 24 STEP 3:GR0,7,0
00060 CURSORX I+1:CURSORY 1:PRINT"WCk"
00065 CURSORX I+1:CURSORY 2:PRINT"U j"
00070 CURSORX I+1:CURSORY 3:PRINT"upz"
00075 CURSORX I+1:CURSORY 4:TX0,0,0
00080 PRINT(I+3)/3:NEXTI:SCREEN:DISPLAY
00090 CURSORX 1:CURSORY 10
00095 PRINT"GIOCATORE ";F+1;
00100 IFF=1THENPRINT" : O":GOTO 110
00105 PRINT" : X"
00110 A$=CHR$(KEY(0))
00115 IFA$<"1"ORA$>"9"GOTO 110
00120 FOR I=1 TO LEN(R$)
00130 IFA$=MID$(R$,I,1)THENGOSUB 170
00140 NEXT I
00150 IFVAL(R$)=0THENCURSORX 1:GOTO 360
00160 GOTO 90
00170 IFF=1THENO$=O$+A$:GOTO 380
00180 IFF=0THENX$=X$+A$:GOTO 400
00190 S$=LEFT$(R$,I-1)+"0"
00195 R$=S$+RIGHT$(R$,LEN(R$)-I)
00200 IFF=1THENT$=O$:GOTO 210
00205 T$=X$
00210 IFLEN(T$)<3GOTO 310
00220 FOR I=1 TO 3
00230 IF I=LEN(T$)GOTO 310
00240 FOR J=I+1 TO LEN(T$)
00250 IFJ=LEN(T$)GOTO 310
00260 FOR K=J+1 TO LEN(T$)
00265 Z=VAL(MID$(T$,I,1))
00270 S=Z+VAL(MID$(T$,J,1))
00275 IF(S+VAL(MID$(T$,K,1)))=15THEN330
00280 NEXT K
00290 NEXT J
00300 NEXT I
00310 F=ABS(F-1)
00320 RETURN
00330 CURSORX 1:CURSORY 10
00340 PRINT"BRAVO , GIOCATORE ";
00345 IFF=1THENPRINT"O":SOUND4:END
00350 PRINT"X":SOUND 4:END
00360 CURSORY 20:PRINT"FINITO":SOUND4
00370 SOUND 4:SOUND 4:END
00380 CURSORX 3*I-1:CURSORY 2:PRINT"O"
00390 GOTO 190
00400 CURSORX 3*I-1:CURSORY 2:PRINT"X"
00410 GOTO 190
```

Destrezza

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessi (1 giocatore).
Difficoltà del gioco: difficile.

IL GIOCO:

Delle cifre, tra 0 e 9, appariranno a caso, in diverse locazioni dello schermo. Restano visualizzate per un tempo molto corto. Il giocatore deve riuscire a battere sul tasto numerico corrispondente, nel momento in cui la cifra appare. La cifra battuta dal giocatore sarà allora visualizzata in grande formato sulla sinistra dello schermo; ed ogni risposta esatta dà 10 punti. La durata della partita è aleatoria ed il punteggio apparirà alla fine della stessa.

IL PROGRAMMA:

Le linee 55 e 65 contengono, in due variabili alfanumeriche C$ e B$, i riferimenti che saranno utilizzati per disegnare le cifre in grande formato.
La posizione della cifra presa dal Videopac è scelta alle linee 70 e 80, e la cifra è presa alla linea 90.
La linea 100 realizza la stampa, e la linea 110 determina la durata della visualizzazione. La linea 130 definisce la durata del gioco, e la fine della partita è trattata alla linea 150.
Il ciclo I (200÷240) realizza il disegno grande formato della cifra battuta dal giocatore.
Il controllo dell'esattezza del risultato è effettuato dalla linea 250.

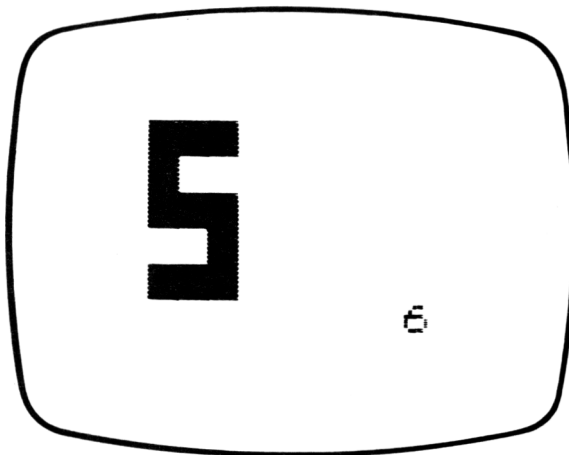
PER GIOCARE:

Il programma sarà lanciato con RUN, poi una prima cifra appare, dopo una corta pausa, verso il mezzo dello schermo. Il giocatore dovrà reagire abbastanza in fretta, intanto che rimane la visualizzazione, e ribatterla.
La cifra che ha battuto si disegnerà allora in alto e a sinistra dello schermo, in grande formato. Se il risultato è esatto, il giocatore segnerà 10 punti. Il tempo di svolgimento della partita dipenderà dalle cifre che saranno visualizzate. Alla fine della partita, il punteggio apparirà. Sarà possibile fermarsi in ogni istante battendo RESTORE.

ESTENSIONI POSSIBILI:

Si può immaginare di modificare il gioco per fare partecipare vari giocatori, modificando eventualmente la visualizzazione e codificando i tasti della tastiera messi a disposizione di ogni giocatore.

Si potranno anche levare dei punti ad ogni risultato sbagliato, modificare le regole di calcolo del punteggio e la durata del gioco, ecc.

```
00010 PRINT"DESTREZZA 0750&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:T=100:S=0:C$=CHR$(127)
00050 K$=C$+C$+C$+C$+"     "
00055 C$=K$+C$+C$+" "+C$
00060 V$="1444133333131211313144133121"
00065 B$=V$+"3112141133331414114131"
00070 X=INT(RND(1)*15)
00080 Y=INT(RND(1)*15+15)
00090 N=INT(RND(1)*10)
00100 CURSORX Y:CURSORY X:PRINTN
00110 FOR I=0 TO RND(1)*750+2000:NEXT I
00120 A$=CHR$(KEY(0)):IFA$<>""THEN 190
00130 T=T-N
00140 CURSORX Y:CURSORY X:PRINT"   "
00150 IFT>=0 THEN70
00160 INIT 7
00170 PRINT:PRINT"Il tuo PUNTEGGIO e'";
00175 PRINT" :";S;" su 100"
00180 SOUND4:SOUND4:SOUND0:SOUND2:END
00190 U=VAL(A$)
00200 FOR I=1 TO 5
00210 A=3*VAL(MID$(B$,5*U+I,1))-2
00220 CURSORX 8:CURSORY 2*I
00230 PRINTMID$(C$,A,3)
00240 NEXT I
00250 IFU=NTHENS=S+10
00260 GOTO 130
```

Sottomarino

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: riflessi (un solo giocatore).
Difficoltà del gioco: facile.

IL GIOCO:

Manovrate in superfice un caccia-sommergibili, e dei sottomarini passano regolarmente sotto la vostra unità, a differenti profondità. Potete, in ogni momento, lanciare le bombe di profondità e potete anche cercare di colpire più volte lo stesso sommergibile. Il vostro punteggio sarà visualizzato in permanenza, sulla sinistra dello schermo (rapporto di numero di colpi centrati su numero di sottomarini apparsi).

IL PROGRAMMA:

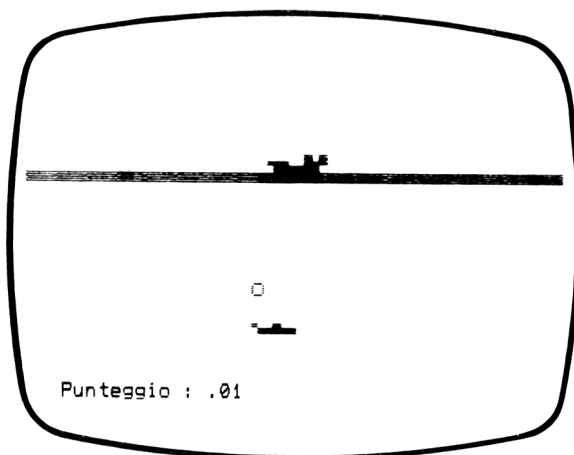
La linea 70 permette al Videopac di scegliere la profondità del sottomarino.
La linea 95 disegna la vostra nave.
Le linee 120 e 115 visualizzano il punteggio e disegnano la superficie dell'acqua.
La linea 110 disegna il sottomarino.
Il lancio è effettuato da un programma specifico posizionato alle linee (200÷240).
Lo spostamento (molto rapido) della vostra nave è realizzato dalle linee 170÷185.
Il test, che permette di sapere se il bersaglio è stato centrato o no, è effettuato alla linea 170.

PER GIOCARE:

Battere RUN per iniziare la partita. La nave sarà spostata grazie alle frecce destra e sinistra (CRSR> o CASA<). Il tasto O autorizza il tiro. Visto che la nave si sposta tre volte più in fretta dei sommergibili, sarà possibile colpire più volte lo stesso sottomarino. Per terminare la partita, converrà battere RESTORE.

ESTENSIONI POSSIBILI:

Sarà abbastanza elementare migliorare la qualità grafica del gioco, utilizzare il colore e il suono, far apparire più sottomarini che si spostano a differenti velocità, eventualmente anche permettere ai sottomarini di lanciare dei siluri...

```
00010 PRINT"SOMMERGIBILE 076@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 P=0:C=15:S=5
00050 INIT 7:PAGE
00060 CURSORX 1:CURSORY S:PRINT"        "
00070 T=35:S=INT(RND(1)*8+4):P=P+.01
00080 X=1
00090 GR0,7,0:STORE:CURSORX C
00095 CURSORY 2:PRINT"@@@ltzpx@@@"
00100 FOR I=1 TO 38:CURSORX I:CURSORY 3
00105 PRINT"?":NEXT I
00110 CURSORX T:CURSORY S:PRINT"dxp@"
00115 CURSORX1:CURSORY18:SCREEN:DISPLAY
00120 TX0,0,0:PRINT"Punteggio :";P
00130 IFX=1THENY=C+2*X
00140 IF(T+1=YORT+2=YORT+3=Y)GOTO 250
00150 T=T-1:IFP=99THENCURSORX1:GOTO 280
00160 IFT<2GOTO 60
00170 A$=CHR$(KEY(0))
00175 K=(A$=CHR$(61)ANDC<27)
00180 Q=(A$=CHR$(47)ANDC>0)
00185 C=C-3*(K-Q)
00190 IFA$<>"0"ANDX=1GOTO 90
00200 CURSORX Y:CURSORY X+1
00210 PRINT" ":X=X+1
00220 IFX>SGOTO 80
00230 CURSORX Y:CURSORY X+1:PRINT"O"
00240 GOTO 90
00250 IF S<>X THEN 150
00255 P=P+1:CURSORXT:CURSORYS
00260 SOUND1:SOUND7:PRINT"***":GOTO 150
00270 CURSORY 18:TX0,3,0:PRINT"AAFFOO";
00280 PRINT"NNDDAATTOO!!":CURSORX1
00290 CURSORY19:PRINT"A F O N D A T ";
00300 PRINT"O ! ":SOUND2:SOUND0:END
```

Zombi

RIFERIMENTI: Difficoltà di programmazione: piuttosto difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: abbastanza facile.

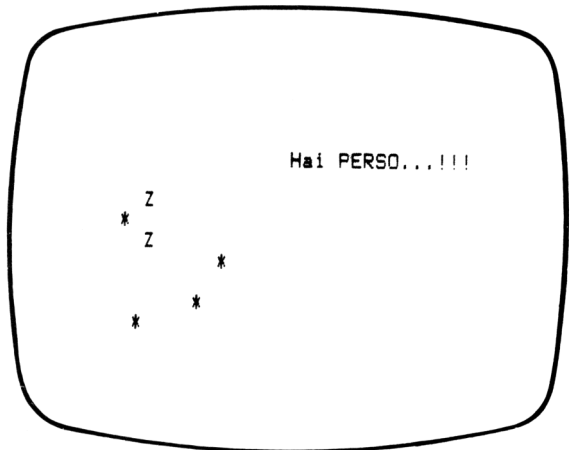
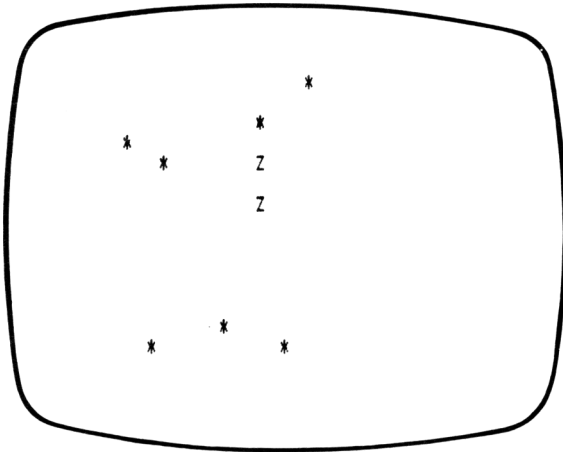
IL GIOCO: Gli ZOMBI sono degli esseri malefici che cercheranno di distruggervi, ma anche voi cercherete di utilizzarli a vostro vantaggio, per sbarazzare il vostro spazio vitale dai buchi neri che l'oscurano. Sullo schermo è disegnato un perimetro che rappresenta il vostro spazio vitale. In questo perimetro, appariranno in maniera aleatoria gli ZOMBI (sono delle Z reverse), i buchi neri (dei quadrati neri) ed il giocatore (una O). Gli Zombi convergono verso il giocatore per cercare di catturarlo, ma lui si sposta più rapidamente di loro, e cercherà di far loro ingoiare i buchi neri, senza farsi prendere. La partita si termina sia con la sparizione del giocatore, sia con l'eliminazione di tutti i buchi neri.

IL PROGRAMMA: La posizione degli ZOMBI è definita da una tabella Z non dimensionata (è inutile per meno di 10).
Le linee 50÷120 (ciclo I) disegnano il perimetro, gli ZOMBI e i buchi neri.
La linea 130 definisce la posizione iniziale del giocatore.
Un secondo ciclo I (150÷240) realizza lo spostamento delle Z.
Le linee (290÷340) controllano lo spostamento del giocatore.

PER GIOCARE: Dopo aver lanciato l'esecuzione con RUN, il giocatore dovrà spostarsi utilizzando le 4 frecce e cercando di non uscire dai limiti del terreno. Dovrà manovrare in modo di fare andare gli ZOMBI sui buchi neri, senza farsi lui stesso bloccare. Per fermare la partita, basterà battere RESTORE.

ESTENSIONI POSSIBILI: Sarebbe interessante ingrandire il campo di gioco, aumentare il numero degli ZOMBI o dei buchi neri, migliorare i controlli (praticamente inesistenti), fare il gioco in maniera più aleatoria, eventualmente modificare la regola (per esempio fare sparire gli ZOMBI nei buchi neri).
Si potrà anche, naturalmente, migliorare la qualità del disegno ed introdurre il colore.

```
00010 PRINT" Z O M B I 077@&"
00020 S$=CHR$(KEY(0)):Y=RND(1)
00030 IFS$=""GOTO 20
00040 INIT 7:Z$=CHR$(127):PAGE:STORE
00050 FOR I=1 TO 18
00060 CURSORX I+1:CURSORY 2:PRINTZ$
00070 CURSORX 1:CURSORY I+1:PRINTZ$
00075 CURSORX 20:CURSORY I+1:PRINTZ$
00080 CURSORX I+1:CURSORY 19:PRINTZ$
00090 IFRND(1)<.5THEN 110
00100 S=INT(RND(1)*15+3)
00105 Z(INT(I/3)+1)=5+S:GOTO 120
00110 CURSORXI+1:Y=INT(RND(1)*15+3)
00115 CURSORY Y:PRINT"*":GOTO 100
00120 NEXT I:SCREEN:DISPLAY
00130 K=INT(RND(1)*17+2)
00140 P=0
00150 FOR I=1 TO 4
00160 P=P+Z(I)
00170 IFZ(I)<3ORZ(I)>18 THEN240
00175 CURSORX Z(I+4):CURSORY Z(I)
00180 PRINT" "
00190 Z(I)=Z(I)-SGN(Z(I)-S)
00200 Z(I+4)=Z(I+4)-SGN(Z(I+4)-K)
00210 CURSORX Z(I+4):CURSORY Z(I)
00220 PRINT"Z"
00230 IFZ(I+4)=KANDZ(I)=STHEN350
00240 NEXT I
00250 IFP=0THENCURSORX 22:GOTO 370
00260 CURSORX K:CURSORY S:PRINT"O"
00270 FOR I=1 TO 1200:NEXT I
00280 CURSORX K:CURSORY S:PRINT" "
00290 A$=CHR$(KEY(0))
00300 Q=2*((A$=CHR$(61))-(A$=CHR$(47)))
00305 K=K-Q
00310 IFK<2ORK>19THENCURSORX22:GOTO390
00320 T=2*((A$=CHR$(45))-(A$=CHR$(43)))
00325 S=S-T
00330 IFS<=2ORS>=19THENK=22:GOTO 310
00340 GOTO 140
00350 CURSORX 22:CURSORY 5
00360 PRINT"Hai PERSO...!!!":GOTO 400
00370 CURSORY10:PRINT"Hai Distrutto "
00380 PRINT"TUTTI gli ZOMBI !!":GOTO400
00390 CURSORY 15:PRINT"PECCATO...!!!"
00400 SOUND0:SOUND2:SOUND4:SOUND5:END
```

Hai PERSO...!!!

Diagonale

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: riflessi (un solo giocatore).
Difficoltà del gioco: difficile.

IL GIOCO:

Si lancia il programma con RUN e si ottiene un primo messaggio che vi informa sulla prestazione da realizzare (per esempio, 5 punti neri, o 17 punti neri...); poi il tempo è visualizzato e decontabilizzato, a partire da 10 secondi. Il gioco comincia quando si arriva a 0 secondi. Dei punti neri appaiono allora in basso a sinistra dello schermo, formando una linea diagonale, e dovrete battere il tasto 0 per fermare la visualizzazione dei punti appena stimerete che il numero di punti richiesti all'inizio è apparso sullo schermo. Si ottiene allora il messaggio BRAVO o PERSO secondo il risultato ottenuto.

IL PROGRAMMA:

Il numero di circoletti sarà scelto a caso, alla linea 50.
Il tempo è decontabilizzato dalle linee 60÷120.
I circoletti sono stampati e la linea diagonale è fabbricata alla linea 140.
L'abbassamento del tasto 0 è accertato alla linea 180 e la linea 190 controlla la fine della partita (nessun riflesso).
I messaggi finali sono eseguiti dalle linee (190÷220).

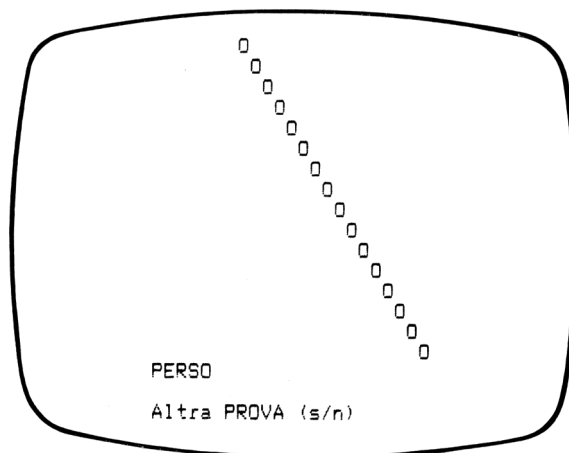
PER GIOCARE:

Si lancerà l'esecuzione con RUN. Si ottiene allora la visualizzazione della frase IL BUON RIFLESSO è A ... che indica il numero di punti vincenti. Il tempo sarà in seguito decontabilizzato sotto forma ATTENZIONE: RESTANO—
——SECONDI che indica il tempo rimanente prima dell'apparizione della diagonale. Alla visualizzazione di 0, la diagonale appare e comincia a spostarsi sullo schermo. Se si stima che il numero di punti richiesti è stato raggiunto, si dovrà premere il tasto 0. Apparirà allora BRAVO se si vince, o PERSO nel caso contrario; poi la frase ALTRA PROVA? (s/n) appare; rispondere s per fare un'altra partita.

ESTENSIONI POSSIBILI:

Si potrebbe pensare ad un adattamento del gioco a due giocatori o più, lasciando la diagonale formarsi completamente e testarne la

coincidenza alla pressione del tasto assegnato al giocatore. Si può anche realizzare un grafismo più curato e modificare il principio dello spostamento degli oggetti sullo schermo (spostamento orizzontale, verticale o apparizione a caso...).

```
00010 PRINT"DIAGONALE 078@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F=Y=RND(1):NEXT I
00040 INIT 7
00050 X=INT(RND(1)*18+3)
00060 S=10
00070 PRINT"Il buon Riflesso e' a ";X
00080 CURSORX1:CURSORY10:PRINT"ATTEN";
00085 PRINT"ZIONE. Restan";S;" Secondi"
00090 S=S-1
00100 SOUND 4:FOR I=1 TO 800:NEXT I
00110 INIT 7
00120 IFS<>0GOTO 70
00130 U=0
00140 CURSORX1:CURSORY20:PRINTTAB(U)"O"
00150 FOR I=1 TO 30:NEXT I
00160 U=U+1
00170 A$=CHR$(KEY(O))
00180 IFA$="0"GOTO 210
00190 IFU>22THENPRINT"PERSO":GOTO 230
00200 GOTO 140
00210 IFU<>XTHENU=23:GOTO 190
00220 PRINT"BRAVO":SOUND0:SOUND4:SOUND2
00230 PRINT:PRINT"Altra PROVA (s/n)"
00240 A$=CHR$(KEY(O)):IFA$="s"GOTO 40
```

Decomaco

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: soluzione del programma Gioco matematico 1 (Nicomaco).
Difficoltà del gioco: nessuna.

IL GIOCO:

È una replica al programma Decomaco, proposto nelle pagine precedenti; inserite i resti successivamente nelle divisioni per 3, 5 e 7 di un intero compreso tra 1 e 100, e il Videopac calcola e visualizza questo intero. Il metodo usato può d'altronde essere generalizzato (a voi trovare come).

IL PROGRAMMA:

Le linee 30-50 visualizzano qualche linea di spiegazione.
La linea 100 aspetta che battiate un resto, che sarà rifiutato se è incompatibile con la divisione col numero visualizzato.
La linea 110 immagazzina i resti in una variabile non dimensionata A (S).
Le linee (125 ÷ 140) rivisualizzano i resti inseriti, e il ciclo (160 ÷ 180) calcola la soluzione ed esegue la stampa.
Le linee (220 ÷ 300) gestiscono la fine della partita.

PER GIOCARE:

Si lancia il programma con RUN. Un testo di presentazione appare, poi il messaggio RESTO DELLA DIVISIONE PER 3 appare. Si deve rispondere 0, 1 o 2, e si ottiene il messaggio RESTO DELLA DIVISIONE PER 5, ecc.
La soluzione è visualizzata appena si sono dati i tre resti.
Per un'altra partita, battere s alla domanda VUOI UN'ALTRA PROVA? (s/n), battere n per finire.

ESTENSIONI POSSIBILI:

Si può estendere il programma agli interi da 1 a 1000, utilizzando i resti per 3, 5, 7, e 11; si possono aggiungere il suono ed il colore per rendere il programma più attraente.

```
00010 PRINT"DECOMACO 079@&"
00020 FOR I=1 TO 100:NEXT I:INIT 7
00025 PAGE:STORE
00030 PRINT"Introduci i 3 RESTI delle"
00040 PRINT:PRINT"DIVISIONI per 3,5,7"
00050 PRINT:PRINT"d'un Num.INTERO <100"
00060 I=3:S=1
00070 CURSORX1:CURSORY8:TX0,1,0:SOUND 4
00075 PRINT"RESTO della DIVISIONE per";
00080 TX0,3,0:PRINTTAB(28)I;" ":TX0,1,0
00085 PRINT"RESTO della DIVISIONE per";
00090 TX0,3,0:PRINTTAB(28)I;" ":TX0,1,0
00095 SCREEN:DISPLAY
00100 A$=CHR$(KEY(0)):IFA$<"0"THEN95
00105 IFA$>CHR$(47+I)THEN95
00110 A(S)=ASC(A$)-48:SOUND 4
00115 S=S+1:I=I+2
00120 STORE:IFI<8THEN70
00125 CURSORX1:CURSORY15:PRINT"RESTO:";
00130 PRINTTAB(9)" per 3  per 5  per 7"
00135 CURSORX1:CURSORY16:PRINT"RESTO:";
00140 PRINTTAB(9)" per 3  per 5  per 7"
00150 TX0,4,0
00160 FOR I=1 TO 3:Z=A(I)
00170 CURSORX 7*I+5:CURSORY 12
00180 PRINTZ;" ":NEXT I
00190 SCREEN:DISPLAY:TX0,1,0
00200 T=A(1)*70+A(2)*21+A(3)*15
00205 IFT>105THENT=T-105:GOTO 205
00210 T$=STR$(T):TS=VAL(LEFT$(T$,2))
00215 TD=VAL(RIGHT$(T$,2)):STORE
00220 CURSORX1:CURSORY18:PRINT"La SOL";
00230 PRINTTAB(6)"UZIONE e'";:TX0,3,0
00240 PRINTSPC(3)TS;" ";TD;" ":TX0,1,0
00250 CURSORX1:CURSORY19:PRINT"La SOL";
00260 PRINTTAB(6)"UZIONE e'";:TX0,3,0
00270 PRINTSPC(3)TS;" ";TD;" "
00275 SCREEN:DISPLAY
00280 TX0,0,0:PRINT:PRINT"Vuoi un'alt";
00290 INPUT"ra PROVA (s,n)";P$
00300 IFP$="s"THENRUN
```

```
Introduci i 3 RESTI delle

DIVISIONI per 3,5,7

d'un Num.INTERO <100

RESTO della DIVISIONE per    7

              2      3      4

RESTO:     per 3  per 5  per 7

La SOLUZIONE e'    5  53
```

Caccia al sottomarino

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco di riflessione.
Difficoltà del gioco: media.

IL GIOCO:

Siete a bordo di una nave, incaricata di scovare un sommergibile nascosto da qualche parte nel vostro settore; dovete spostarvi fino ad arrivare sopra al sottomarino e sparare ad una certa profondità; mà, attenzione: il sommergibile è anch'esso mobile; e spegnerà i suoi motori solo quando sarete esattamente sulla verticale, per non essere scoperto.

IL PROGRAMMA:

Il ciclo I (60÷120) visualizza il tavolo di gioco.
La linea 140 definisce la posizione iniziale del sottomarino; la linea 150 permette il suo spostamento, controllato dalla linea 160 (si resta nei limiti del campo).
La linea 180 visualizza la vostra posizione.
Le linee (220÷245) controllano il vostro spostamento.
La distanza è calcolata alla linea 250, visualizzata alla linea 265 e testata alle linee 270 e 280.
Il test di fine partita è effettuato alla linea 310.

PER GIOCARE:

Si lancia il programma con RUN e si inserisce la dimensione; la griglia di gioco appare. Siete all'inizio alla posizione (0,0) e vi spostate con le frecce. Dopo ogni spostamento, la nuova distanza del sottomarino appare; ad ogni colpo il sottomarino si sposta. Se riuscite a piazzarvi sulla verticale, la distanza 0 appare e non si muove più. Sparate dando una profondità intera (tra 2 e 4, limiti inclusi). Se lo mancate, si riposiziona in un posto qualunque, e si ricomincia; se lo colpite, ottenete il messaggio BRAVO e la partita si termina. Si può fermare il programma in ogni momento utilizzando il tasto RESTORE.

ESTENSIONI POSSIBILI:

È possibile prevedere una miglioria del grafico, migliorare le possibilità di spostamento del caccia (un po' lento, in questa versione), prevedere dei percorsi più sofisticati per il sottomarino, di mettere un contatore di tempo...

```
00010 PRINT"Caccia-SOTTOMARINO 080@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TOF:Y=RND(1):NEXT I:PRINT
00040 INPUT "Dimensione GRIGLIA :";H
00050 IFH>7GOTO 40
00055 INIT 7:PAGE:STORE
00060 FOR I=0 TO H
00070 CURSORX 2*I+3:CURSORY 2:PRINTI
00080 CURSORX 1:CURSORY 2*I+4:PRINTI
00090 FOR J=0 TO H
00100 CURSORX 2*J+4:CURSORY 2*I+4
00105 PRINT"."
00110 NEXT J
00120 NEXT I:SCREEN:DISPLAY
00130 I=1:X=0:Y=0
00140 V=INT(RND(1)*H):U=INT(RND(1)*H)
00150 U=U+I:V=V+I
00160 IF(U<0ORU>HORV<0ORV>H)THEN I=-I
00170 CURSORX 2*Y+2:CURSORY 2*X+4
00175 IFY<>0THENPRINT"."
00180 CURSORX 2*Y+4:CURSORY 2*X+4
00185 PRINTCHR$(127)
00190 CURSORX 2*Y+6:CURSORY 2*X+4
00195 IFY<>HTHENPRINT"."
00200 CURSORX 2*Y+4:CURSORY 2*X+2
00205 IFX<>0THENPRINT"."
00210 CURSORX 2*Y+4:CURSORY 2*X+6
00215 IFX<>HTHENPRINT"."
00220 A$=CHR$(KEY(0)):IFA$=""THEN 220
00230 A=Y-(A$=CHR$(61)ANDY<H)
00235 Y=A+(A$=CHR$(47)ANDY>0)
00240 B=X-(A$=CHR$(45)ANDX<H)
00245 X=B+(A$=CHR$(43)ANDX>0)
00250 D=INT(SQR(ABS(X-V) 2+ABS(Y-U) 2))
00260 CURSORX 24:CURSORY 10
00265 PRINT"DISTANZA :";STR$(D)+" ";
00270 IFD=0THENSOUND 2:GOTO 300
00280 IFD<3THENSOUND 4:GOTO 150
00290 GOTO 170
00300 CURSORX 24:CURSORY 12
00305 INPUT "PROFONDITA' :";D
00310 IFD=INT(RND(1)*3+2)THEN335
00320 CURSORX 24:CURSORY 12
00325 PRINT"                  ":SOUND 0
00330 GOTO 140
00335 PRINT"BRAVO...!!!":SOUND1:SOUND2:
00340 END
```

```
  0 1 2 3 4 5 6 7

0 . .   . . . . .

1 . . . . . . . .

2 . . . . . . . .

3 . . . . . . . .     DISTANZA : 0

4 . . . . . . . .     PROFONDITA' :? 4

5 . . . . . . . .

6 . . . . . . . .

7 . . . . . . . .
```

Tastiera

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: gioco di riflessi (imparare la tastiera).
Difficoltà del gioco: difficile.

IL GIOCO:

Una riproduzione della tastiera del vostro calcolatore appare sullo schermo; un punto appare sotto, e la colonna nella quale si trova il punto indica il carattere corrispondente, che conviene di battere sulla tastiera. Più si batte in fretta, e più il punto si sposta rapidamente (il tempo trascorso dall'inizio appare in fine partita).

IL PROGRAMMA:

Una variabile A$ dimensionata alla linea 60, conterrà la lista dei tasti utilizzati.
I valori dei tasti sono inseriti dalla linea 70 alla linea 120, per richiamo di un sottoprogramma situato in 300.
I cicli chiusi J (130÷170) e I (140-160) disegnano la tastiera.
La linea 180 definisce la posizione dell'asterisco, stampato dalla linea 195.
Il tasto battuto è testato alle linee 220, 230 e l'asterisco è cancellato dalla linea 255.
Il numero di tentativi è totalizzato alla linea 240.
I messaggi finali sono realizzati alle linee 275 e 280.

PER GIOCARE:

Si lancia il programma con RUN. Si dispone allora di qualche secondo per battere il tasto che occupa la colonna sopra la pallina, poi un'altra colonna è visualizzata, ecc.
La durata della partita non è limitata; basta battere 0 per ottenere il punteggio, ed il tempo totale. Fare RUN per un'altra partita.

ESTENSIONI POSSIBILI:

Si potrà pensare di fare lampeggiare le lettere, o di visualizzarle a colori, al posto di indicare la colonna con una pallina. Si potranno fare apparire le lettere giuste nel basso dello schermo, e eventualmente allenarsi a battere un testo che sfila in alto sullo schermo, lettera per lettera.

```
00010 PRINT"TASTIERA 081@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 T=0:P=0:S=0
00060 DIMA$(10,3)
00070 S$="QWERTYUIOP":J=1
00080 GOSUB 300
00090 S$="ASDFGHJKL":J=2
00100 GOSUB 300
00110 S$="ZXCVBNM.?":J=3
00120 GOSUB 300
00130 FOR J=1 TO 3
00140 FOR I=1 TO 10
00150 CURSORX 3*I+J:CURSORY 5*J:TX0,0,0
00160 PRINTA$(I,J)
00170 NEXT I:NEXT J
00175 X=INT(RND(1)*10+1)
00180 Y=INT(RND(1)*3+1)
00190 CURSORX 3*X+Y:CURSORY 2+5*Y
00195 PRINT"*"
00200 FOR I=1 TO 50*RND(1)*80:NEXT I
00205 A$=CHR$(KEY(0)):T=T+1
00210 IFA$=""THEN200
00220 IFA$=A$(X,Y)THENS=S+1
00230 IFA$="0"THEN270
00240 P=P+1
00250 CURSORX3*X+Y:CURSORY2+5*Y
00255 PRINT" "
00260 GOTO 175
00270 CURSORX1:CURSORY20:TX0,0,0
00275 PRINTS;" GIUSTI su  ";P;"Prove"
00280 PRINT"Il tuo TEMPO :";T
00290 SOUND 4:SOUND 5:SOUND 2:END
00300 FORI=1TO10:A$(I,J)=MID$(S$,I,1)
00310 NEXT I:RETURN
```

```
Q W E R T Y U I O P

 A S D F G H J K L

  Z X C V B N M . ?
```

La lucertola di Alice

RIFERIMENTI:

Difficoltà di programmazione: facile.
Categoria del gioco: solitario.
Difficoltà del gioco: piuttosto difficile.

IL GIOCO:

Si ispira assai vagamente a un'idea di Lewis Carrol nel suo "Alice nel Paese delle Meraviglie". Nel racconto originale Alice ingrandiva o diminuiva, mentre nel gioco, per limitazioni delle dimensioni di memoria, è una lucertola che cambia le sue dimensioni.
Sullo schermo compaiono due bicchieri: uno contenente il liquido che fa ingrandire, l'altro quello che rimpicciolisce (sfortunatamente non si tratta sempre dello stesso bicchiere). Il giocatore dovrà indicare la quantità di liquido che desidera bere e la lucertola cambierà le sue dimensioni in funzione di questa quantità, ma non linearmente. L'obiettivo del gioco è di farle riprendere la dimensione iniziale.

IL PROGRAMMA:

Il ciclo I (60÷90) disegna i bicchieri. I due cicli chiusi I (190÷220) e J (195÷210) li riempiono con le quantità indicate alla linea 175.
La variazione delle dimensioni della lucertola viene definita dalla linea 160 e controllata dalle linee 270 e 280.
Il ciclo I (290÷310) disegna il corpo della lucertola e le linee 320 e 370 disegnano l'occhio e le zampe. Un messaggio di vincita viene visualizzato dalla linea 395 se si è trovata la dimensione giusta.

PER GIOCARE:

Si lancia il programma con RUN.: i due bicchieri vengono disegnati, poi appare la domanda "Che LATO", cui si risponde battendo "s" o "d"; appare allora la domanda "Che QUANTITÀ"; si introduce allora una quantità da 0 a 80 e si batte RET. Il bicchiere si riempie sotto i vostri occhi poi, dopo qualche istante, la lucertola compare sullo schermo. Si dovrà allora battere un tasto qualsiasi; se la lucertola non aveva la dimensione giusta, non appare nessun messaggio e si ritorna al disegno dei bicchieri.

ESTENSIONI POSSIBILI:

Trattandosi di un gioco essenzialmente grafico, sarà interessante migliorare i disegni, introdurre i colori, modificare il carattere casuale della scelta dei bicchieri, ecc.
Perché non rileggere "Alice nel Paese delle Meraviglie"? Le idee non mancano!

```
00010 PRINT"Lucertola d'ALICE * 082@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE:STORE
00050 N=15:Z$=CHR$(127)
00060 FOR I=1 TO 9
00070 CURSORX  3:CURSORY I+4:PRINTZ$
00075 CURSORX 10:CURSORY I+4:PRINTZ$
00080 CURSORX 13:CURSORY I+4:PRINTZ$
00085 CURSORX 20:CURSORY I+4:PRINTZ$
00090 NEXT I
00100 FOR I=3 TO 10
00105 CURSORXI:CURSORY 14:PRINTZ$:NEXTI
00110 FOR I=13 TO 20
00115 CURSORXI:CURSORY 14:PRINTZ$:NEXTI
00120 CURSORX4:CURSORY15:PRINT"SINIST."
00125 CURSORX14:CURSORY15:PRINT"DESTRO"
00130 CURSORX1:CURSORY17:SCREEN:DISPLAY
00135 PRINT"Che LATO (d/s) ? "
00140 A$=CHR$(KEY(0)):IF A$="d" THEN150
00145 IF A$<>"d" AND A$<>"s" THEN 140
00150 U=3-10*(A$="d")
00160 CURSORX20:CURSORY17
00165 IFA$="d"THENPRINT"DESTRO":GOTO175
00170 PRINT"SINISTRO"
00175 PRINT:INPUT "Che QUANTITA' :";Q
00180 IFQ>80THEN175
00190 FOR I=1 TO SQR(Q)
00195 FOR J=1 TO 6
00200 CURSORX U+J:CURSORY 14-I
00210 PRINT"$":NEXT J
00220 NEXT I:SOUND 2
00230 B$=CHR$(KEY(0)):IF B$="" THEN 230
00240 STORE
00250 H=2*(RND(1)<.5)-1
00260 N=N+H*INT(SQR(Q))
00270 IFN<3THENN=3
00280 IFN>16THENN=16
00290 FOR I=1 TO N+3
00300 CURSORX I:CURSORY 2:PRINT"%"
00310 NEXT I
00320 T=2*(N-INT((SQR(N)/2)))
00330 CURSORX T:CURSORY 2:TX0,4,0
00340 PRINT"o":T=2*SQR(N)+1:TX0,0,0
00350 CURSORX T:CURSORY 3:PRINT"1"
00360 T=2*(N-SQR(N))
00370 CURSORX T:CURSORY 3:PRINT"1"
00375 SCREEN:DISPLAY
00380 C$=CHR$(KEY(0)):FOR I=1 TO 100
00385 IF C$="" THENNEXT I:J=I:GOTO 380
00390 NEXT I:CURSORY 25:TX0,4,1
00395 IFN=3THENPRINT"BRAVO!":SOUND2:END
00400 IFJ<100 THEN TX0,0,0:GOTO 40
00410 PRINT"Che BARBA! GIOCO Io per TE"
00420 FOR I=1 TO 600:NEXT I
00430 Q=INT(RND(1)*100+10)
00440 GOTO 240
```

Calcolatrice

RIFERIMENTI:

Difficoltà di programmazione: molto difficile.
Categoria del gioco: simulazione di una calcolatrice.
Difficoltà del gioco: media.

IL GIOCO:

Si tratta della simulazione, con visualizzazione in grande formato, del funzionamento di una piccola calcolatrice tascabile, del tipo a 4 operazioni. Le applicazioni pedagogiche di questo programma sono evidenti e il suo funzionamento è semplice da controllare.

IL PROGRAMMA:

Le variabili B$ e C$, delle linee 25 e 35, contengono le informazioni codificate che permetteranno il disegno delle cifre grande formato.
Le linee 60-150 si occupano della raffigurazione, in grande formato, delle cifre sullo schermo, man mano che gli ingressi arrivano.
Il punto decimale ed il segno — sono gestiti dalle linee 200 e 210.
Il numero totale dei simboli inseriti è controllato dalla linea 220.
Il risultato è iscritto grazie ad un programma funzionante automaticamente che si trova alle linee 230 e 235 (l'automatismo è gestito dalle linee 270÷390, nel programma principale).
I sottoprogrammi posizionati alle linee 400÷410, 420÷430, 440÷450 e 460÷470 effettuano l'operazione indicata, secondo il valore di Z (linea 310). Ogni risultato è memorizzato in A (F-1) (linea 330).

PER GIOCARE:

Si dovrà lanciare il programma con RUN. Poi, si battono le cifre del primo numero da inserire che appariranno una per una, man mano (non si potranno inserire più di 8 simboli, compreso il segno — ed il punto decimale). La battuta di un segno aleatorio lascia la visualizzazione intatta, ma l'introduzione del prossimo simbolo o cifra provoca la cancellazione immediata e la visualizzazione del nuovo simbolo. È da notare che le cifre si introducono da sinistra a destra, senso usuale di scrittura. L'apparecchio accetta, come segni operatori i simboli +,—,*,/ ; il segno = provocherà l'apparizione del risultato. Le operazioni sono concatenate automaticamente, il risultato precedente è preso in considerazione come operatore del calcolo seguente. Per fare un'altra operazione, bisognerà battere RESTORE, poi

RUN. Battere RESTORE per uscire dal programma.

ESTENSIONI POSSIBILI: Si dovrà, innanzitutto, migliorare la visualizzazione del segno — (raffigurato qui da un punto), poi si potrà migliorare il disegno delle cifre (utilizzando, per esempio, il generatore del Videopac); si può anche pensare ad un'estensione a 8 cifre, autorizzare delle operazioni più complesse (COS, SIN,...), aggiungere delle memorie, ecc.).

```
00010 PRINT"CALCOLATRICE 083@&"
00015 C$=CHR$(127):CLEAR1000
00020 C$=CHR$(127):G$=C$+C$+C$+C$
00025 C$=G$+"    "+C$+C$+" "+C$
00030 P$="1444133333131211313144133 12"
00035 B$=P$+"1311214113333141411413 1"
00040 F=1:L=0:INIT 7:TX0,1,0
00050 N=0:T=0:K=0:R$=""
00060 A$=CHR$(KEY(0)):IFA$=""THEN60
00065 IF A$=CHR$(0) THEN 60
00070 IFK=1THENN=0:R$="":K=0
00080 IFA$="="THEN260
00090 X=(1ANDA$="+")+(2ANDA$="-")
00095 X=X+(3ANDA$="*")+(4ANDA$="/")
00100 R$=R$+A$
00110 IFX<>0 THEN260
00120 IFR$="-"THENA$=R$:GOTO 200
00130 IFA$<"."ORA$>"9"ORA$="/"THEN60
00140 IFN=0THENINIT 7
00150 IFA$="."ORA$="-"THEN200
00160 FOR I=1 TO 5
00170 S=3*VAL(MID$(B$,I+5*VAL(A$),1))-2
00180 CURSORX N+1:CURSORY I+3:
00185 PRINTMID$(C$,S,3)
00190 NEXT I
00200 IFA$="-"THENGOSUB 480
00210 IFA$="."THENGOSUB 500
00220 N=N-4*(N<24)
00230 IFT=0THEN240
00235 T=T+1:IFT< LEN(R$)THEN380
00240 IFK=1ANDL<>1THEN50
00250 T=0:GOTO 60
00260 IFR$="-"ANDA(F)=0THEN120
00270 N=0:K=1
00280 IF(A$="*"ORA$="/")GOTO 520
00290 A(F)=VAL(R$)
00300 IFX<>0THENZ=X
00310 ONZGOSUB 400,420,440,460
00320 IFA$="="THENF=2:GOTO 330
00325 F=F+1
00330 A(F-1)=VAL(R$)
00340 IFVAL(R$)=0THENA(F-1)=A(F)
00350 R$=STR$(VAL(R$))
00360 IFLEFT$(R$,1)<>" "THEN370
00365 R$=RIGHT$(R$,LEN(R$)-1)
00370 T=1
00380 A$=MID$(R$,T,1)
00390 GOTO 140
00400 R$=STR$(A(F-1)+A(F))
00410 A(F)=VAL(R$):L=1:RETURN
00420 IFF=1THENRETURN
00425 R$=STR$(A(F-1)-A(F)):A(F)=VAL(R$)
00430 L=1:RETURN
00440 IFF=1THENRETURN
00445 R$=STR$(A(F-1)*A(F)):A(F)=VAL(R$)
00450 L=1:RETURN
00460 IFF=1THENRETURN
00465 R$=STR$(A(F-1)/A(F)):A(F)=VAL(R$)
00470 L=1:RETURN
00480 INIT 7:TX0,3,0
00485 CURSORX 1:CURSORY 7:PRINTA$;A$
00490 CURSORX 1:CURSORY 8:PRINTA$;" "
00495 TX0,1,0:RETURN
00500 TX0,3,0
00505 CURSORXN+2:CURSORY 7:PRINT".";" "
00510 CURSORXN+2:CURSORY 8:PRINT".";"."
00515 TX0,1,0:RETURN
00520 IFVAL(R$)=0THENA(F)=1:GOTO 300
00530 GOTO 290
```

LIVELLO 5

N.	Nome del programma	Tema e particolarità
84	Smokey	Fumate senza rischio con questo originale programma per un solo giocatore.
85	Corsa automobilistica	Gioco di riflessi non troppo difficile, per un giocatore.
86	Scacchi	Programma molto semplificato, per due giocatori.
87	Pericolo spaziale	Sarete così in gamba da evitare tutte le meteoriti?
88	(Tic Tac Toe) - Tris	Un classico, ma non senza interesse, per due giocatori.
89	Grafica speciale	Per evitarvi il tedio di codificare i caratteri speciali!
90	Dama	Un classico per due giocatori.
91	Emulazione	Per trasformare una stampante in una macchina per scrivere.
92	Anagrammi	Un buon gioco di riflessi, per due giocatori.
93	Uova	Non date tregua all'"ovaiola diabolica", per un solo giocatore.
94	Puzzle di lettere	Vi è nascosto un proverbio: provate a scoprirlo.
95	Bersagli	Un po' di destrezza e molta fortuna. Molto visivo.
96	Gocce	Un vero e proprio effetto visivo che si può sfruttare!
97	Ostacoli	Molti ostacoli da evitare in questo gioco d'abilità per un giocatore.
98	Catture nello spazio	La vostra missione sarebbe facile senza quei dannati meteoriti.
99	Anomalia	Esercitate la vostra perspicacia nel riconoscere i simboli.
100	Tanti in uno	L'ambiente di gioco ed i pedoni, per due giocatori.
101	Salti di cavallo	Assai spettacolare ed istruttivo da più d'un punto di vista.
102	Muro di mattoni	Un gran classico in una versione piuttosto completa.
103	Othello	Per due giocatori; riporta i pedoni al vostro posto.

L'iscrizione e la lettura diretta nella memoria

Nel Basic Microsoft fornito all'origine sul VIDEOPAC+ non esiste un'istruzione diretta che consenta la lettura della memoria dedicata allo schermo, la quale occupa (2 * 40 * 23 = 1840) posizioni della RAM (Memoria ad Accesso casuale), a partire dal 32776 al 34615.
In molti giochi si ha la necessità di riconoscere il codice ASCII del carattere presente sullo schermo all'intersezione della Colonna X e della Riga Y. Si può procedere in due modi:

— Si costruisce una mappa dello schermo mediante una tabella che viene aggiornata man mano che la visualizzazione avviene, adoperando le seguenti istruzioni:

DIM VD(40,22)

VD(X,Y) = F(m,n) che rappresenta il risultato visualizzato

IF VD(X,Y) = CHAR$ (Codice ASCII) THEN...

che effettua un test di presenza o meno del carattere che interessa verificare perché condiziona lo sviluppo successivo del programma di gioco. Questa procedura è lenta e piuttosto onerosa in termini di linee di programma e quindi di tempi d'esecuzione.
Conviene ricorrere ad un sistema più elegante che è vantaggioso per la rapidità d'esecuzione oltre che per la ridotta occupazione di memoria. In fondo costruirsi una mappa di memoria ausiliaria è un ripiego e un inutile doppione se si ha il modo d'accedere direttamente alle memorie riservate allocate a questo scopo dal Basic Microsoft, basta fare ricorso alle istruzioni riportate sul Manuale d'Uso del VIDEOPAC+ per leggere e/o scrivere direttamente nella memoria, esse sono:

— La funzione PEEK che permette la lettura diretta del contenuto di una delle locazioni di memoria assegnabili dal microprocessore.
— La funzione POKE che permette l'iscrizione diretta di una quantità (da 0 a 255 in decimale) in una di queste locazioni di memoria.

Le due funzioni PEEK e POKE sono complementari fra loro e sono utili operando in linguaggio assemblativo, per caricare o richiamare sub-routine, per scambiare argomenti e risultati, ecc.
Ritornando al problema di lettura dello schermo video, basterà fare:

PEEK (2 * X + 80 * Y −32760)

per sapere ciò che si trova sullo schermo alle coordinate X ed Y, ottenendo un numero, compreso fra 0 e 255, che corrisponde al codice del carattere presente nella memoria dello schermo.
Abbiamo perciò utilizzato questa seconda soluzione, che non solo è più brillante e più veloce oltre a far risparmiare memoria, tenuto conto anche del fatto che il VIDEOPAC+ dispone soltanto di due tabelle dimensionabili con DIM. L'esempio in cui s'è usata la prima procedura chiarirà le ragioni di tale preferenza.
La funzione POKE è stata utilizzata in collegamento con l'estensione TURBO del VIDEOPAC+ per attivare la stampante seriale che ci è servita per la stampa delle liste di istruzioni qui riprodotte, con il comando:

POKE −30760, 40

Ovviamente volendo accelerare lo svolgimento dei giochi, in particolar modo di quelli d'animazione, sarebbe opportuno usare delle sub-routine in linguaggio assemblativo o in codice macchina, le quali possono essere richiamate tramite la funzione USR, ma si tratta di programmazioni avanzate che esulano dagli scopi di questo libro.

Smokey

RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO: Una colonna di fumo grigio appare sullo schermo. Il giocatore, rappresentato da un punto nero, diventa visibile sulla destra. Deve in un primo tempo posizionarsi sulla colonna di fumo, poi, partendo da lì, ingoierà automaticamente il fumo, a patto che riesca a mantenersi sulla traccia.

IL PROGRAMMA: Le linee 90 e 95 fanno apparire il fumo nel basso dello schermo; le linee 100 e 105 permettono di cambiare (verso destra o verso sinistra) la direzione della colonna, e la linea 120 fa salire la colonna.
Il giocatore appare grazie alle linee 140 e 145, ed il suo spostamento è effettuato dalle linee 180 e 190.
Le linee 250 e 260 si incaricano della visualizzazione del risultato.

PER GIOCARE: Si deve lanciare l'esecuzione del programma con RUN. Il fumo comincia ad alzarsi, mentre il giocatore, apparso nella forma di un puntino colorato in un angolo dello schermo, si dovrà spostare con l'aiuto delle 4 frecce verso la colonna. Appena arriva sul fumo, comincerà automaticamente ad ingoiarlo, e dovrà cercare di seguire la colonna senza deviare (grazie alle frecce 8 e 5). Vince se arriva nel basso dello schermo prima che la colonna arrivi in alto. Il messaggio finale indica allora BRAVO o PERSO, così come il punteggio ottenuto.

ESTENSIONI POSSIBILI: Si potrà, conservando l'idea di partenza, imporre delle altre regole: per esempio ingoiare tutto il fumo, che potrà anche ripartire da un altro posto se si devia la colonna iniziale. Si potranno anche migliorare i controlli di fine partita e la qualità del grafismo.

```
00010 PRINT"S M O K E Y 084@&"
00020 INPUT "Dammi un Num.INTERO :":F
00030 FORI=1TOF:Y=RND(1):NEXT I
00040 SETEG32,"001C3E7F7F7f3E1C0000"
00050 S=0:X=20
00060 Y=INT(RND(1)*30)+3
00070 U=12:V=15
00080 INIT 7:PAGE
00090 CURSORX Y:CURSORY X
00095 PRINTCHR$(24)
00100 Z=Y-(RND(1)<.5ANDY<20)
00105 Y=Z+(RND(1)<.5ANDY>0)
00110 T=1
00120 X=X-T
00130 IFX=0THEN245
00140 CURSORX V:CURSORY U:EG0,7,0
00145 PRINTCHR$(32):TX0,0,0
00150 IFT=0GOTO 170
00160 A$=CHR$(KEY(0)):IFA$=""THEN160
00170 TX0,0,0
00175 CURSORXV:CURSORYU:PRINT" "
00180 U=U-(A$=CHR$(45)ANDT=1ANDX<20)
00190 U=U-(T=0)
00200 B=V-(A$=CHR$(61)ANDV<20)
00205 V=B+(A$=CHR$(47)ANDV>0)
00210 C=PEEK(2*V+80*U-32760)
00220 IFCHR$(C)<>CHR$(24)THEN90
00230 T=0:S=S+1
00240 IFU<19THENEG0,7,0:GOTO 140
00245 TX0,0,0
00250 IFT=0THENPRINT"BRAVO";:GOTO 270
00260 TX0,4,0:PRINT"PERSO...":SOUND 5
00270 PRINT"PUNTEGGIO :";S:END
```

Corsa automobilistica

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco di riflessi (un giocatore).
Difficoltà del gioco: media.

IL GIOCO:

Guidate su una strada molto sinuosa, senza nessun ostacolo, e dovete cercare di seguirla per il maggior tempo possibile. La traccia del vostro percorso resta visualizzata in permanenza, e il vostro punteggio appare subito dopo un incidente.

IL PROGRAMMA:

La stampa della strada è assicurata dal ciclo I (70-100) e la linea 110 stampa il colore.
La linea 100 fa avanzare la strada, e la linea 120 stampa la macchina.
La linea 150 controlla il suo movimento, e la linea 160 determina se la strada fa o no delle curve.
La linea 170 guarda se abbiamo toccato il bordo della strada.
La linea 180 incrementa il punteggio, e la linea 190 cancella la vecchia posizione della macchina (una traccia chiara appare nella posizione della sua traiettoria).
La linea 220 visualizza il tempo fino all'incidente.

PER GIOCARE:

Lanciare l'esecuzione del programma con RUN, il disegno della strada comincia ad evolversi, ed il vostro veicolo, rappresentato da un cuore appare.
Per spostarlo verso destra o verso sinistra, dovrete utilizzare le frecce di spostamento del cursore (verso destra e sinistra). Dovrete evitare per il maggior tempo possibile di toccare il bordo nero, il che provocherà l'arresto del programma e la visualizzazione del messaggio INCIDENTE, seguito dall'indicazione del punteggio realizzato. Si potrà fermare lo svolgimento della partita in ogni istante col tasto RESTORE.

ESTENSIONI POSSIBILI:

Si può immaginare di giocare in molti, ognuno al suo turno, registrando il miglior punteggio. Si può anche pensare di allargare la pista, e di fare correre varie macchine. Potremo anche ispirarci ai giochi dei bar che sono particolarmente numerosi su questo tema.

```
00010 PRINT"CORSA d'Automobili 085@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FORI=1TOF:Y=RND(1):NEXT I
00040 SETEG35,"5D7F55325D7F5D080000"
00050 T=0:INIT 7:Z$=CHR$(127)
00060 X=19:Y=11:A=10:B=15:S=0:STORE
00070 FOR I=1 TO 7
00080 CURSORX Y+I:CURSORY X:PRINTZ$
00090 IFI<2ORI>4THENGOSUB 240
00100 NEXT I:PRINT:N=N+1
00110 EG0,7,0:SCREEN:DISPLAY
00120 CURSORX B:CURSORY A:PRINTCHR$(35)
00130 R=R+1:A$=CHR$(KEY(0)):T=T+.1
00135 IFA$=""ANDR<=10THEN130
00140 C=B
00150 B=B-(A$=CHR$(61))+(A$=CHR$(47))
00155 W=Y-(2*RND(1)ANDY<30)
00160 Y=W+(2*RND(1)ANDY>3)
00165 IFN<9THENTX0,0,0:GOTO 70
00170 K=PEEK(2*(B-1)+80*(A+1)-32760)
00171 TX0,0,0:PRINTK:EG0,7,0
00175 IFK<>127 THENTX0,0,0:GOTO 210
00180 S=S+1:TX0,0,0
00190 CURSORX C:CURSORY A:PRINT" "
00200 GOTO 70
00210 PRINT"INCIDENTE-Punteggio :";S
00220 PRINT:PRINT"TEMPO :";INT(T)
00230 SOUND 4:SOUND 0:SOUND 4:END
00240 CURSORX Y+I:CURSORY X:PRINTZ$
00250 RETURN
```

Scacchi

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco di riflessione, per due giocatori.
Difficoltà del gioco: difficile.

IL GIOCO:

Il programma propone solamente ai due giocatori di fare una partita servendosi dello schermo come scacchiera. Non c'è controllo della validità delle mosse, ma solamente la gestione della posizione degli scacchi sulla scacchiera. Le pedine e gli altri pezzi sono disegnati con caratteri grafici SETEG (41÷46). Il colore (verde o giallo) permette di distinguere ogni mossa.

IL PROGRAMMA:

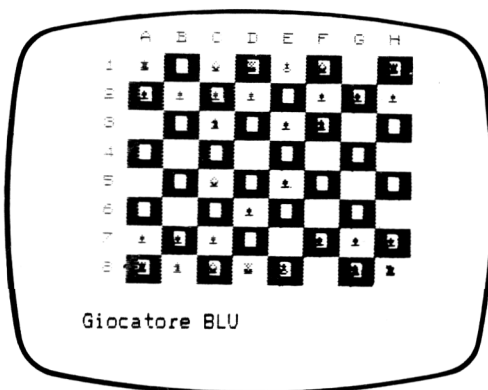
Le linee 130÷230 disegnano la scacchiera.
Le linee (235÷250) definiscono le condizioni iniziali, con l'aiuto di due tabelle; R(I,J) per i codici dei pezzi, e C(I,J) per il colore.
I cicli chiusi J (260÷295) e I (265÷290) disegnano gli scacchi.
La linea 350 permette l'ingresso della linea e della colonna di partenza, col controllo alle linee (345÷355).
La linea 380 permette di inserire la colonna e la linea di arrivo.
La linea 400 nota la nuova posizione.
La linea 410 assicura il cambio di giocatore.

PER GIOCARE:

Si lancia l'esecuzione con RUN. La scacchiera è disegnata, poi i pezzi, ed infine il messaggio GIOCATORE BLU, seguito da PARTENZA (C e L); si inseriscono la colonna e la linea di partenza, nella maniera E7, per esempio, si ottiene allora il messaggio ARRIVO (C e L); e si risponde nello stesso modo.
Lo spostamento è allora effettuato, e sarà il turno dell'altro giocatore.
Per finire, basta battere RESTORE.

ESTENSIONI POSSIBILI:

Si potrà cercare di realizzare un grafismo più accurato, introdurre i controlli di movimento dei pezzi. E perché no, scrivere un programma che permetta al Videopac di comportarsi da buon partner. Buon lavoro!

```
00010 PRINT" SCACCHI 0860&"
00020 DIMR(8,8):DIMC(8,8)
00030 O$=CHR$(127)+CHR$(127)+CHR$(127)
00040 SETEG40,"001028446C387C000000"
00050 SETEG41,"001038103844380000000"
00060 SETEG42,"00547C38387C7C000000"
00070 SETEG43,"005428106c7c7c000000"
00075 SETEG44,"0000103838107C000000"
00080 SETEG45,"0010387818387c000000"
00090 SETEG46,"00000000000000000000"
00100 INIT7:STORE:CURSORX5:CURSORY1
00110 PRINT"A  B  C  D  E  F  G  H"
00120 PRINT
00130 FOR I=1 TO 8
00140 PRINT:PRINTI
00150 FOR J=1 TO 8
00160 IF(I+J)/2=INT((I+J)/2)GOTO 205
00170 CURSORX 3*J+1:CURSORY 2*I+1
00180 PRINTO$
00190 CURSORX 3*J+1:CURSORY 2*I+2
00200 PRINTO$
00205 IFI=2ORI=7THENR(I,J)=44
00210 IFI<>2ANDI<>7THENR(I,J)=46
00215 C(I,J)=1
00220 IFI=7ORI=8THENC(I,J)=46
00225 NEXT J
00230 NEXT I:SCREEN:DISPLAY
00235 R(1,1)=42:R(1,2)=45:R(1,3)=40
00240 R(1,4)=43:R(1,5)=41:R(1,6)=40
00245 R(1,7)=45:R(1,8)=42
00250 FOR I=1 TO 8:R(8,I)=R(1,I):NEXT I
00255 H=1
00260 FOR J=1 TO 8
00265 FOR I=1 TO 8
00270 X=3*J+2:Y=2*I+2
00275 CURSORX X:CURSORY Y
00280 EG1,7,0:IFH=1THENEG4,7,0
00285 PRINTCHR$(R(I,J))
00290 NEXT I
00295 NEXT J:TX0,0,0:PAGE
00300 CURSORX34:CURSORY19:PRINT"        "
00305 CURSORX25:CURSORY20:IFH=1THEN530
00310 TX1,4,0:PRINT"Giocatore ROSSO"
00315 CURSORX1:CURSORY21:TX0,0,0
00320 PRINT"                    "
00325 CURSORX1:CURSORY21
00330 INPUT"PARTENZA C&L:";D$
00335 CURSORX34:CURSORY19:PRINTD$
00340 GOSUB 500
00345 IFR(L,C)=46THEN305
00350 IFC(L,C)=41ANDH=1THEN305
00355 IFC(L,C)=46ANDH=0THEN305
00360 T=R(L,C):S=C(L,C):R(L,C)=46
00365 CURSORX1:CURSORY21:TX0,0,0
00370 PRINT"                    "
00375 CURSORX1:CURSORY21
00380 INPUT "ARRIVO C&L:";D$
00385 CURSORX36:CURSORY19:PRINT"-";D$
00390 GOSUB 500
00400 R(L,C)=T:C(L,C)=S
00410 H=ABS(H-1)
00420 GOTO 260
00500 L=VAL(RIGHT$(D$,1))
00510 C=ASC(LEFT$(D$,1))-64
00520 RETURN
00530 TX4,4,0:PRINT"Giocatore BLU  "
00540 GOTO 315
```

Pericolo spaziale

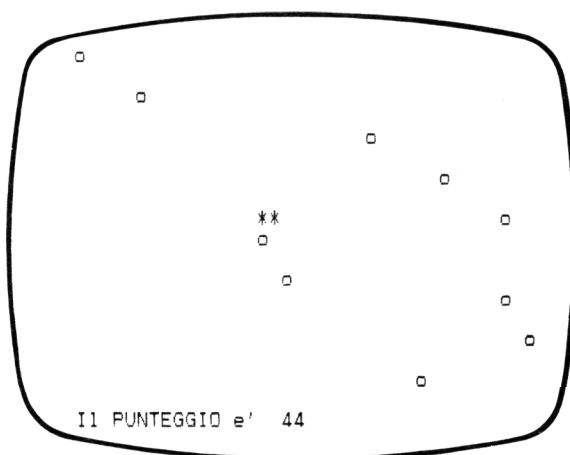
RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: gioco di riflessi, per un giocatore.
Difficoltà del gioco: media.

IL GIOCO: Siete ai comandi di una nave spaziale, che è appena penetrata in uno sciame di meteoriti. Dovete cercare di uscire dallo sciame senza urtare nessuna meteorite. il vostro punteggio sarà visualizzato alla fine della partita, quando colliderete con una meteorite.

IL PROGRAMMA: La posizione di ogni meteorite è determinata dalle linee 60 e 80; e sono stampate dalle linee 200÷210.
L'avanzamento delle meteoriti è realizzato dalla linea 100.
Le linee 100 e 110 verificano se avete o no urtato una meteorite.
Le linee 130-135 disegnano la vostra astronave, e le linee 130 e 135 permettono di spostarla.
Il contatore si trova alla linea 170.

PER GIOCARE: Il gioco è lanciato con l'ordine RUN. La vostra astronave appare nel mezzo dello schermo, e le meteoriti vi vengono incontro partendo dal basso dello schermo. Per spostarvi, potete utilizzare le frecce di spostamento orizzontale. Si può fermare la partita in ogni istante battendo RESTORE. Altrimenti il gioco si ferma quando urtate una meteorite, e il punteggio appare.

ESTENSIONI POSSIBILI: Sarebbe interessante rendere il gioco sempre più difficile, facendo apparire un numero sempre maggiore di meteoriti man mano che la partita si svolge.
Si può anche cercare di distruggere le meteoriti polverizzandole man mano che arrivano.

```
00010 PRINT"PERICOLO Spaziale 087@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 X=18:Z=1:Y=X
00060 H=INT(RND(1)*38+1)
00065 TX1,0,0:CURSORY 20
00070 IF RND(1)<.7 THEN GOSUB 200
00080 K=INT(RND(1)*38+1)
00090 IF RND(1)<.6 THEN GOSUB 210
00100 PRINT:B=PEEK(2*X+80*11-32760)
00105 C=PEEK(2*(X+1)+80*11-32760)
00110 IF B=111 OR C=111 THEN190
00120 CURSORX Y-1:CURSORY10:PRINT"     "
00125 CURSORX Y-1:CURSORY 9:PRINT"     "
00130 CURSORX X:CURSORY 12:TX6,0,0
00135 PRINT"**"
00140 A$=CHR$(KEY(0))
00150 Y=X
00160 X=X-(A$=CHR$(61)ANDX<38)
00165 X=X+(A$=CHR$(47)ANDX>2)
00170 Z=Z+1
00180 GOTO 60
00190 TX0,4,0:PRINT"Il PUNTEGGIO e' ";Z
00195 TX0,0,0:SOUND 2:END
00200 CURSORX H:PRINTCHR$(111):RETURN
00210 CURSORX K:PRINTCHR$(111):RETURN
```

(Tic Tac Toe) - Tris

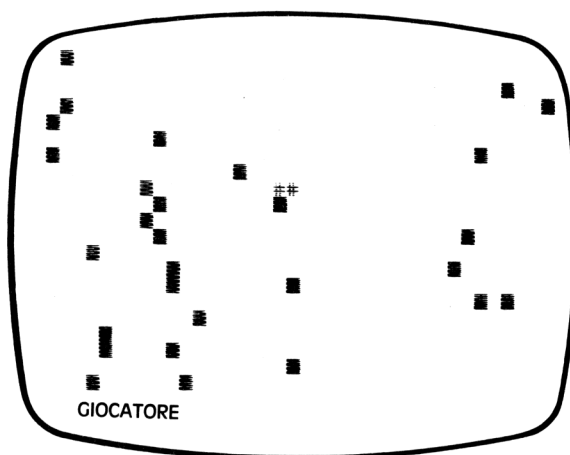
RIFERIMENTI: Difficoltà di programmazione: media.
Categoria del gioco: riflessione (due giocatori).
Difficoltà del gioco: facile.

IL GIOCO: Un grande classico dei giochi "scolastici", proposto qui in una versione che non comporta il gioco da parte del calcolatore; ogni giocatore dovrà posizionare nel suo turno uno 0 o una X in una casella libera della griglia di gioco, e il primo che riesce ad allineare tre simboli identici ha vinto (l'allineamento può essere orizzontale, verticale, o diagonale).

IL PROGRAMMA: La griglia di gioco è disegnata con l'aiuto dei cicli I (45÷80) e J (50÷70).
La linea 110 si occupa di visualizzare il simbolo cuore o picca corrispondente al giocatore che ha la mano.
Un simbolo lampeggiante (linee 140 e 145) è visualizzato in alto e a sinistra della griglia, anche se la casella è già occupata; le linee (160÷175) permettono di spostare il simbolo nella griglia.
La linea 180 verifica se la casella dove si vuole giocare è libera, e se si desidera giocare.
La linea 190 visualizza la mossa, e la linea 210 realizza il cambio del giocatore.

PER GIOCARE: La griglia di gioco è visualizzata dalla battuta di RUN. Il simbolo lampeggiante appare allora in alto a sinistra dello schermo; il giocatore può spostarlo nella griglia con l'aiuto delle 4 frecce. Quando arriva sulla casella che desidera, basta che prema il tasto 0, la mossa sarà rifiutata se la casella non è libera. Se la casella era libera, il simbolo corrispondente a quel giocatore sarà iscritto, e si cambia di giocatore. Quando la partita è finita, conviene battere RESTORE per uscire dal programma. Si potrà in seguito fare un'altra partita con RUN.

ESTENSIONI POSSIBILI: Si potrebbe realizzare dopo ogni colpo un controllo per verificare se uno dei giocatori ha vinto. Si può anche realizzare un gioco per il calcolatore ma bisognerebbe evitare che sia troppo sistematico (per esempio, non giocare sempre la prima mossa sulla casella centrale).

```
00010 PRINT"TRIS(Tic-Tac-Toe) 088@&"
00020 SETEG32,"00367F7F3E1C08000000"
00030 SETEG33,"00081C3E7F3E1C080000"
00035 S$=CHR$(127):Z$=S$+S$+S$+S$+S$+S$
00040 INIT 7:PAGE:STORE
00045 FOR I=1 TO 3
00050 FOR J=1 TO 12
00060 CURSORX4*I-2:CURSORYJ+2:PRINTZ$
00070 NEXT J
00080 NEXT I:K=33:SCREEN:DISPLAY
00090 CURSORX 1:CURSORY 18:TX0,2,0
00100 PRINT"G I O C A T O R E :   ":
00110 EG0,7,0:PRINTCHR$(K):I=1:J=1
00115 C=PEEK(2*(J+1)+80*(I+1)-30760)
00120 IFC=127THENK=32
00125 TX0,3,0:CURSORX 4*I:CURSORY 4*J
00130 PRINT"X  ":CURSORX4*I:CURSORY4*J+1
00135 PRINT"X  ":TX0,0,0
00140 IFC<>127THENPRINTCHR$(C):GOTO 155
00150 TX0,0,0:PRINTCHR$(127)
00155 A$=CHR$(KEY(0))
00160 G=J-(A$=CHR$(45)ANDJ<3)
00165 J=G+(A$=CHR$(43)ANDJ>1)
00170 E=I-(A$=CHR$(61)ANDI<3)
00175 I=E+(A$=CHR$(47)ANDI>1)
00180 IFC=127ORA$<>"0"THEN115
00190 CURSORX 4*J:CURSORY 4*I
00195 EG0,7,0:PRINTCHR$(K)
00200 K=ABS(K-1)
00210 IFK=1THENSOUND 4:SOUND 6:GOTO 90
00220 SOUND 2:SOUND 3:GOTO 90
```

Grafica speciale

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: utilità.
Difficoltà del gioco: facile.

IL GIOCO:

Scopo del programma è mettere a disposizione dell'utente un codificatore esadecimale di caratteri speciali del Videopac+, evitando la noiosa procedura che occorre seguire disegnando su carta quadrettata il carattere da definire. A questo modo la creazione di caratteri speciali diventa un divertimento anziché un perditempo.

IL PROGRAMMA:

Il sottoprogramma (450÷480) provvede all'azzeramento della matrice dei caratteri in apertura di programma.
I cicli I (40÷80) e J (50÷80) visualizzano i punti del carattere, mentre il ciclo I (90÷110) visualizza i codici esadecimali corrispondenti.
Le linee (120÷150) controllano il cursore lampeggiante nonché i punti (rettangolini neri) che formano il carattere prescelto.
Le linee (160÷210) consentono la modifica del carattere e ne controllano i movimenti. La linea 220 verifica se la penna è stata abbassata, ossia se il punto della matrice dev'essere bianco o nero. In caso positivo, le linee (230÷380) calcolano i codici esadecimali relativi al punto in oggetto e predispongono LA LORO STAMPA.
La linea 380 esegue il test di completamento del carattere, rimandando eventualmente al programma stampa iniziale delle linee 120 e 130.
Il sottoprogramma dalla linea 500 alla fine disegna il riquadro.

PER GIOCARE:

Si lancia il programma con RUN attendendo che venga evidenziato sullo schermo il tracciato del riquadro, che ha il cursore al centro. Usando i tasti con le frecce si realizza lo spostamento del cursore. Per abbassare la penna, cioè per annerire una posizione occorre premere il tasto (*) simbolizzato anche dalla freccia inclinata a sinistra verso l'alto. Viceversa premendo il tasto (*) su una posizione nera la si cancella. Terminato il tracciato, basta battere (0) per terminare il programma e stabilizzare l'elenco dei codici esadecimali.

ESTENSIONI POSSIBILI: Per completare il programma si può sviluppare il problema inverso che, dati i codici esadecimali, esegue nel riquadro il disegno del carattere corrispondente. Per chi possiede il joy-stick può essere interessante trasferire a quest'ultimo i comandi degli spostamenti e della penna.

```
00010 PRINT"GRAFICA SPECIALE 0890&"
00020 DIM K(9,7):GOSUB 450
00030 INIT 3:STORE:Z$=CHR$(127):G=4:L=3
00040 FOR I=0 TO 9
00050 FOR J=0 TO 7
00060 CURSORX 10+J:CURSORY 5+I
00070 PRINTCHR$(32+95*K(I,J))
00080 NEXT J:NEXT I:GOSUB 500
00090 FOR I=1 TO 20 STEP 2
00100 CURSORX 21:CURSORY 5+(I-1)/2
00110 PRINT MID$(C$,I,2):NEXT I:SCREEN
00120 CURSORX 10+L:CURSORY 5+G:TX2,0.1
00130 PRINTZ$::SCREEN:TX0,0,0
00140 CURSORX 10+L:CURSORY 5+G
00150 PRINTCHR$(32+95*K(G,L)):TX2.0,1
00160 G=G-(KEY(0)=45)+(KEY(0)=43)
00170 L=L-(KEY(0)=61)+(KEY(0)=47)
00180 IF G<0 THEN G=0
00190 IF G>9 THEN G=9
00200 IF L<0 THEN L=0
00210 IF L>7 THEN L=7
00220 IF KEY(0)=42 THEN 230
00225 GOTO 390
00230 K(G,L)=1-K(G,L):I=G*2+INT(L/4)+1
00240 C1$=LEFT$(C$,I-1)
00250 C3$=RIGHT$(C$,20-I)
00260 C=ASC(MID$(C$,I.1))-48
00270 IF C>9 THEN C=C-39
00280 C1=L:IF L>3 THEN C1=C1-4
00290 C1=3-C1:IF K(G,L)=1 THEN 310
00300 C=C AND(255-2 C1):GOTO 320
00310 C=C OR 2 C1
00320 IF C<10 THEN 340
00330 C2$=CHR$(87+C):GOTO 350
00340 C2$=RIGHT$(STR$(C),1)
00350 C$=C1$+C2$+C3$
00360 CURSORX 10+L:CURSORY 5+G:TX0.0,0
00370 PRINTCHR$(32+95*K(G,L));
00380 CURSORX 21:PRINTMID$(C$,2*G+1,2)
00390 IF KEY(0)<>48 THEN 120
00400 DISPLAY:TX0.0,1:PRINT"FINITO":END
00450 FOR I=0 TO 9
00460 FOR J=0 TO 7
00470 K(I,J)=0:NEXT J:NEXT I
00480 C$="00000000000000000000":RETURN
00500 CURSORX 10:CURSORY 4
00510 PRINT"84218421":TX7.0,0
00520 FOR I=4 TO 15
00530 CURSORX 9:CURSORY I:PRINTZ$
00540 CURSORX18:CURSORY I:PRINTZ$
00550 NEXT I
00560 CURSORX 9:CURSORY 4:DELIM7.7.0
00570 CURSORX19:CURSORY 4:DELIM3,3.0
00580 CURSORX 9:CURSORY15:DELIM7.7.0
00590 CURSORX19:CURSORY15:DELIM3.3.0
00600 TX 0,0.0:SCREEN:SOUND 4:RETURN
```

Dama

RIFERIMENTI:

Difficoltà di programmazione: molto difficile.
Categoria del gioco: due giocatori.
Difficoltà del gioco: difficile.

IL GIOCO:

È il gioco classico della dama alla francese, ma un po' incompleto visto che non prevede la possibilità di fare delle dame. La regola del gioco resta classica, ma i controlli sono stati un po' semplificati. I movimenti delle pedine sono inseriti in una sola manovra (posizione di partenza e direzione dello spostamento) e le prese sono realizzate automaticamente, se sono possibili. Invece il cambio del giocatore è realizzato ad ogni mossa, quindi bisognerà eventualmente battere 0 per poter rigiocare se c'era stato un salto o se il colpo non era possibile.

IL PROGRAMMA:

I cicli I (70÷200) e J (90÷190) disegnano la scacchiera e definiscono le posizioni iniziali delle pedine nere e di quelle bianche. Le posizioni sono in una tabella R (X,Y) ed il colore in un'altra tabella C (X,Y). La stampa delle pedine vera e propria è effettuata dai cicli chiusi J (220÷290) e I (230÷280).
Le linee 320 e 330 indicano il giocatore che ha il gioco in mano.
La mossa è inserita alla linea 350. La linea 350 realizza il cambio del giocatore (si batte 0).
La linea 300 guarda se lo spostamento è possibile. La linea 420 tratta il caso di uno spostamento e la linea 430 quello di un salto.
La linea 490 assicura il cambio del giocatore.

PER GIOCARE:

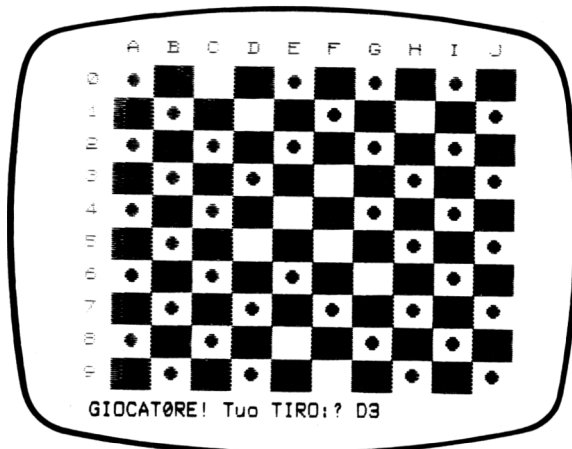
Dopo il lancio del programma con RUN, la scacchiera appare con le sue pedine, e un carattere o reverse appare sotto per indicare che la prima mossa tocca al giocatore che ha scelto i bianchi. Si gioca indicando con un numero di 3 cifre (che può cominciare con uno 0) la linea e la colonna della pedina da spostare, poi la direzione dello spostamento (codificata da 1 a 4) secondo la convenzione

4 1
3 2

Per esempio 931 significa che la O situata alla linea 9, colonna 3, si sposta nella direzione 1 (non dimenticare di fare RETURN). Nella stessa maniera, 063 significherà pedina nera, linea 0, colonna 6, direzione 3.

Il cambio del giocatore essendo effettuato automaticamente, bisognerà battere 0 per rigiocare se la mossa è stata rifiutata o se c'è stata una presa (terminare sempre con RETURN). Il programma non controlla l'arrivo a dama, e non proibisce gli spostamenti indietro.

ESTENSIONI POSSIBILI: Sarà abbastanza facile cominciare col migliorare i controlli (cambio di giocatore, mossa indietro che non corrisponderebbe a una presa, uscita dalla scacchiera). Si potrà pensare alla creazione di dame, il loro spostamento, ed eventualmente scrivere un programma che permetterebbe al Videopac di giocare con voi, con buone possibilità di vincita.

```
00010 PRINT" D A M A  Francese 0900&"
00020 DIMR(10,10):DIMC(10,10)
00030 O$=CHR$(127)+CHR$(127)+CHR$(127)
00040 SETEG32,"00000000000000000000"
00050 SETEG33,"001C3E7F7F7F3E1C0000"
00055 INIT 7:PAGE:CURSORX5:CURSORY0
00060 PRINT"A  B  C  D  E  F  G  H  I";
00065 PRINT"  J":STORE
00070 FOR I=1 TO 10
00080 PRINTI-1:PRINT
00090 FOR J=1 TO 10
00100 IF(I+J)/2=INT((I+J)/2)GOTO 150
00110 CURSORX 3*J+1:CURSORY 2*I-1
00120 PRINTO$
00130 CURSORX 3*J+1:CURSORY 2*I
00140 PRINTO$
00150 R(I,J)=32
00160 IF(I+J)/2<>INT((I+J)/2)THEN190
00170 IFJ<5THENR(I,J)=33:C(I,J)=0
00180 IFJ>6THENR(I,J)=33:C(I,J)=6
00190 NEXT J
00200 NEXT I:SCREEN:DISPLAY
00210 H=1:STORE
00220 FOR J=1 TO 10
00230 FOR I=1 TO 10
00240 IF(I+J)/2<>INT((I+J)/2)THEN280
00250 X=3*J+2:Y=2*I
00260 CURSORXX:CURSORYY:EGC(I,J),7,0
00270 PRINTCHR$(R(I,J))
00280 NEXT I
00290 NEXT J:SCREEN:DISPLAY
00300 CURSORX22:CURSORY21:PRINT"        "
00310 CURSORX1:CURSORY21
00320 TX0-6*(H=1),0,0:PRINT"GIOCAT0RE";
00330 EG0-6*(H=1),7,0:PRINTCHR$(33);
00340 TX0-6*(H=1),0,0:PRINT" Tuo TIRO";
00345 INPUT ":";D$
00350 IFD$="0"THENH=ABS(H-1):GOTO 300
00360 GOSUB 500
00370 IFR(L,C)=32THEN300
00380 IF(C(L,C)=0ANDH=1)THEN300
00390 IF(C(L,C)=6ANDH=0)THEN300
00400 T=(1AND(D=1ORD=2))
00405 T=T-(1AND(D=4ORD=3))
00410 S=(1AND(D=3ORD=2))
00415 S=S-(1AND(D=1ORD=4))
00420 IFR(L+S,C+T)=32THEN540
00430 IFC(L+S,C+T)=0-6*(H=1)THEN570
00450 GOTO 300
00460 R(L,C)=32:R(L+S,C+T)=32
00470 R(L+2*S,C+2*T)=33
00480 C(L+2*S,C+2*T)=0-6*(H=1):GOTO220
00490 H=ABS(H-1):GOTO 220
00500 L=VAL(LEFT$(D$,1))+1
00510 C=ASC(MID$(D$,2,1))-64
00520 L=VAL(RIGHT$(D$,1))
00530 RETURN
00540 R(L,C)=32:R(L+S,C+T)=33
00560 C(L+S,C+T)=0-6*(H=1):GOTO 490
00570 IFR(L+S,C+T)=33GOTO 590
00580 GOTO 300
00590 IFR(L+2*S,C+2*T)=32THEN460
```

Emulazione

RIFERIMENTI:

Difficoltà di programmazione: media.
Categoria del gioco: utilità.
Difficoltà del gioco: nessuna.

IL GIOCO:

Non si tratta qui di un gioco ma di un doppio programma che consente di emulare un sistema di scrittura "word processing" in tempo reale, e di ottenere la copia stampata del contenuto dello schermo, ovviamente se avete l'estensione "TURBO" del Videopac.
Potrete così crearvi un archivio personale, scrivere le vostre lettere, stampare le vostre circolari, o semplicemente stampare un elenco d'indirizzi. Che fatica in meno l'invio degli auguri per le prossime feste.
Il programma s'accontenta di esplorare la memoria di schermo, linea per linea, e di trasformare i codici video in codici stampa. (Attenzione: non tutti i codici sono equivalenti, dovete conoscere la vostra stampante!).

IL PROGRAMMA:

Innanzitutto il programma è sistemato nelle ultime posizioni di memoria disponibili per l'utilizzatore (65500÷65529), che raramente vengono adoperate.
In linea 65501 si definisce una tabella di 39*22, quante sono le posizioni disponibili sullo schermo del Videopac. I cicli I (65504÷65511) e J (65505÷65511) consentono la cattura del tasto battuto correlandolo alla sua posizione sullo schermo, e danno il comando di "a capo". I cicli I (65513÷65516) e J (65514÷65515) visualizzano sulla stampante le eventuali pagine di schermo archiviate.
Le linee 65520÷65529 riguardano il programma di stampa vero e proprio, che può avere un'utilizzazione autonoma in molte applicazioni che siamo certi adopererete ampiamente.
Le linee (65521÷65523) squadrano lo schermo e sono utili per centrare la pagina stampata, se vi occorre, mentre i cicli I (65525÷65529) e J (65526÷65529) eseguono la stampa riga per riga e posizione per posizione. L'istruzione avanzata POKE è usata per l'esplorazione della memoria di schermo.

PER GIOCARE:

Inserito il programma, vi sarà data la scelta se archiviare o no il testo che v'accingete a battere, poi potete procedere come su una normale macchina per scrivere.

Se invece volete semplicemente listare il vostro programma o stampare il contenuto dello schermo per avere, ad esempio, la copia scritta dei giochi che vi proponiamo, caricate "EMULAZIONE" in coda al vostro programma con una "CLOAD A 65520" dopo esservi accertati che ciò non interferisca con le posizioni di memoria occupate dal vostro programma. Fatti gli opportuni rimandi, al termine del gioco, se non vi sono caratteri grafici e/o speciali, il gioco è fatto ed otterrete la documentazione stampata della vostra bravura o della vostra sfortuna!
Si utilizzerà il programma sia prevedendo un GOTO o GOSUB in posizione opportuna nel programma principale, sia fermando il programma con un BREAK e scrivendo GOTO o GOSUB (numero di linea di EMULAZIONE che v'interessa, max 65525) direttamente nel basso dello schermo.

ESTENSIONI POSSIBILI: Il vostro programma "word processing" potrà essere migliorato, con una più raffinata gestione dei codici di carattere, estendendolo ai caratteri grafici e a quelli speciali. Si può prevedere la correzione della battuta, la stampa estesa a tutta la lunghezza della riga disponibile sulla vostra stampante (80, 120 e 136 caratteri), il tempo di risposta della vostra battuta potrà essere adeguato alle vostre capacità dattilografiche, ecc...

```
65500 REM-TYPING Emulation on TURBO/vcg
65501 INIT7:CLEAR5000:DIMM$(39,22):PAGE
65502 CURSORY22:INPUT"COPIA(1)";S:INIT7
65503 CURSORY 0:PAGE:IFS=1THEN65513
65504 FOR I=0 TO 21
65505 FOR J=1 TO 39
65506 T$=CHR$(KEY(0)):FOR Y=1TO2:NEXTY
65507 IFT$=CHR$(0)THEN65506
65508 M$(J,I)=T$
65509 IFT$="!"THENJ=39:SOUND2:GOTO65511
65510 IFT$<>CHR$(0)THENPRINTT$;:SOUND 4
65511 NEXT J:PRINT:NEXT I:SOUND 2
65512 INIT 7:PAGE:CURSORY 0
65513 FOR I=0 TO 21:STORE
65514 FOR J=1 TO 39
65515 PRINTM$(J,I);:NEXT J
65516 SCREEN:DISPLAY:PRINT:NEXT I
65517 CURSORY22:PRINT"Stampa(RET):";
65518 PAGE:INPUT S$:S=0:GOSUB 65524
65519 S=0:GOTO 65502"
65520 REM-Screen COPY on G7420TURBO/vcg
65521 PAGE:SOUND 2:CURSORY0:PRINT".";
65522 PRINTTAB(39);".":CURSORY 22
65523 PRINT".";TAB(39);".":S=1
65524 CURSORY 1-S
65525 FOR I=0 TO 21+S
65526 FOR J=1 TO 39
65527 C=PEEK(2*J+80*I-32760)
65528 LPRINTTAB(J);CHR$(C);
65529 NEXT J:LPRINT:NEXT I:RETURN
```

Anagrammi

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco di riflessi (2 giocatori).
Difficoltà del gioco: secondo il livello.

IL GIOCO:

Delle lettere sono scelte a caso dal Videopac, il numero di lettere essendo funzione del livello di difficoltà desiderato, poi le lettere della parola così ottenuta sono mescolate, e visualizzate nel basso dello schermo; una nuova parola è allora creata, in modo aleatorio, con le stesse lettere, ed è visualizzata sotto la prima, e così via; i due giocatori (destro e sinistro) dispongono ognuno di un tasto (1 o 0) e devono premerlo appena vedono visualizzarsi in basso una parola già apparsa; il loro punteggio è allora calcolato, ed un messaggio corrispondente al loro risultato apparirà allora.

IL PROGRAMMA:

Il livello di gioco è scelto alla linea 50.
Il ciclo I (80÷100) realizza la scelta delle lettere.
Un secondo ciclo I (120-190) effettua la mescola, e la linea 140 permette di accertare la battuta di un tasto.
La linea 200 visualizza una nuova parola, e fa salire di una linea sullo schermo le parole già ottenute.
La linea 245 cerca quale dei due giocatori ha premuto il tasto.
Il ciclo I (250-320) esplora le parole presenti sullo schermo linea per linea, mentre il ciclo J (270-300) prende una per una le lettere di queste parole.
Si ottiene così una nuova parola che è paragonata alla parola iniziale alla linea 310 ed il risultato BRAVO appare se è esatto.
Altrimenti, la partita si ferma sul messaggio PERSO (linea 340).

PER GIOCARE:

Si lancia il programma con RUN. Si dovrà anche inserire il livello di difficoltà (in teoria da 1 a 3) e fare RETURN. Le parole si visualizzano partendo dal basso dello schermo, e salgono man mano; se uno dei giocatori crede di riconoscere nella parola apparsa una delle parole già visualizzate, dovrà immediatamente premere il suo tasto (1 o 0), ed il suo risultato sarà visualizzato dopo un istante. La partita sarà allora terminata.

ESTENSIONI POSSIBILI: Si potranno utilizzare non solo le parole, modificare i tempi di visualizzazione, fare partecipare più di due giocatori, introdurre il suono ed il colore, ecc.

```
00010 PRINT"A N A G R A M M I 0920&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7
00050 INPUT "LIVELLO (da 1 a 3) :";N
00060 IFN<1ORN>3THEN40
00070 S$=""
00080 FOR I=1 TO N+3
00090 S$=S$+CHR$(INT(RND(1)*26+65))
00100 NEXT I
00110 A$="":B$=S$
00120 FOR I=1 TO LEN(S$)
00130 N=INT(RND(1)*LEN(S$)+1)
00140 D$=CHR$(KEY(0))
00150 IFD$="9"ORD$="0"THEN240
00160 IFMID$(B$,N,1)="-"THEN130
00170 A$=A$+MID$(B$,N,1)
00180 C$=LEFT$(B$,N-1)+"-"
00185 B$=C$+RIGHT$(B$,LEN(B$)-N)
00190 NEXT I
00200 CURSORY 21:PRINTA$
00210 FOR I=1 TO 20*LEN(B$):NEXT I
00220 S$=A$
00230 GOTO 110
00240 H$="Giocatore SINISTRO"
00245 IFD$="9"THENH$="Giocatore DESTRO"
00250 FOR I=0 TO 22
00260 B$=""
00270 FOR J=1 TO LEN(S$)-1
00280 R$=CHR$(PEEK(2*J+80*I-32760))
00290 B$=B$+R$
00300 NEXT J
00310 IFB$=S$THEN350
00320 NEXT I
00330 CURSORX LEN(S$)+4:CURSORY 21
00340 PRINT"PERSO ";H$::SOUND 0:END
00350 CURSORX LEN(S$)+4:CURSORY I
00360 PRINT"BRAVO , ";H$:SOUND 2:END
```

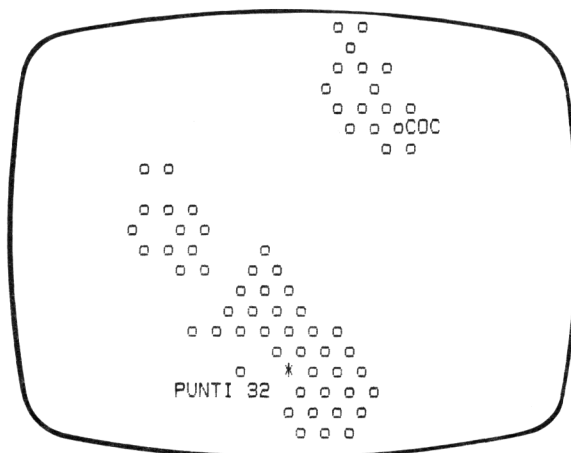
```
TVANIL
IATVNL
ANITVL
VATLNI
ANVLTI
INALVT
NLIVTA
ITALVN
NVATLI
VLTINA
ATLIVN
VLANTI
LVITNA
LNAIVT
NVITLA
VINTLA
NLIAVT
TINLVA
IVLTNA
LTAINV
AVNLIT      PERSO Giocatore DESTRO
```

Uova

RIFERIMENTI:	Difficoltà di programmazione: difficile. Categoria del gioco: solitario. Difficoltà del gioco: media.
IL GIOCO:	Un uccello misterioso si sposta, battendo le ali, a caso sullo schermo. Pone delle uova (raffigurate da delle O) che voi (un quadrato grigio) dovrete cercare di recuperare. Il vostro punteggio (numero di uova recuperate) sarà visualizzato in permanenza, e la partita si svolge in un tempo determinato.
IL PROGRAMMA:	L'uccello è disegnato in due parti (per il movimento) alla linea 40; il suo spostamento aleatorio è generato dal sottoprogramma (210÷300) (in particolare le linee 260÷290). La linea 130 permette di riciclare il programma sulla linea 70. Lo spostamento del giocatore è gestito dalle linee (150÷165), ed il test che permette di accertare la presenza di un uovo è effettuato dalla linea 170. La linea 190 permette la visualizzazione del punteggio. La fine del gioco è effettuata alla linea 320.
PER GIOCARE:	Si lancerà il programma con RUN. L'uccello comincia allora a spostarsi a caso sullo schermo, e si mette a deporre le uova. Quando il simbolo che rappresenta il giocatore (un quadrato grigio) appare sullo schermo, costui può spostarsi utilizzando i 4 tasti di spostamento. Si deve allora cercare di recuperare un massimo di uova prima della fine della partita. Si potrà fermare il gioco in qualsiasi momento battendo RESTORE.
ESTENSIONI POSSIBILI:	Sullo stesso principio, potremo facilmente immaginare degli altri temi di gioco. Potremo naturalmente introdurre dei suoni, i colori, migliorare il grafismo, ed eventualmente, definire un gioco con vari giocatori, ognuno che cercherebbe di recuperare un massimo di uova.

```
00010 PRINT"  U O V A  093@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00035 INIT 7:PAGE
00040 S$="COC":Z$="LOL"
00050 P=200:R=18:S=10:T=0
00055 X=INT(RND(1)*34+5)
00060 Y=INT(RND(1)*21)
00070 CURSORX X:CURSORY Y:GR0,7,0
00075 PRINTZ$:TX0,0,0
00080 A$=CHR$(KEY(0))
00090 IFA$<>CHR$(0)THENGOSUB 140
00100 GOSUB 210
00110 CURSORX X:CURSORY Y:GR0,7,0
00115 PRINTS$:TX0,0,0
00120 GOSUB 210
00130 GOTO 70
00140 CURSORX R:CURSORY S:TX0,0,0
00145 PRINT" "
00150 L=R-(A$=CHR$(61)ANDR<39)
00155 R=L+(A$=CHR$(47)ANDR>1)
00160 Z=S-(A$=CHR$(45)ANDS<22)
00165 S=Z+(A$=CHR$(43)ANDS>0)
00170 C=PEEK(2*R+80*S-32760)
00180 IFC=111THENT=T+1:SOUND 4
00190 CURSORX9:CURSORY19:PRINT"PUNTI";T
00200 RETURN
00210 A=X:B=Y
00220 P=P-1
00230 IFP=0THENSOUND 2:END
00240 CURSORXR:CURSORYS:TX0,4,0
00245 PRINT"*":TX0,0,0
00250 CURSORX A:CURSORY B:PRINT" o "
00260 X=ABS(X+INT(1+2*(RND(1)>.5)))
00270 Y=ABS(Y+INT(1+2*(RND(1)<.5)))
00280 IFX>38ORX<1THENX=1
00290 IFY>21THENY=1
00300 RETURN
```

oCOC

PUNTI 32

"Puzzle" di lettere

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: difficile.

IL GIOCO:

Chi di noi non ha, un giorno o l'altro, avuto tra le mani uno di quei piccoli puzzles dove si cerca di spostare da una casella all'altra dei piccoli quadrati di plastica colorata, fino ad ottenere un risultato ben preciso? Al suo tempo lo scacciaquindici, di SAM LOYD, ha avuto lo stesso successo del celebre CUBO di RUBIK.
Il puzzle che è proposto qui utilizza 35 lettere, disposte a caso in un quadrato di 6*6 caselle, e queste lettere possono, riposizionate nel loro ordine iniziale, permettervi di ricostituire una frase. Una casella grigia vi permette di spostare le lettere, ed un contatore visualizza in permanenza il numero di tentativi: riuscirete a trovare la frase?

IL PROGRAMMA:

La frase da ricostituire (che potrà naturalmente essere modificata) è posizionata in una variabile R$ alle linee 320 e 330.
La mescola delle lettere e la stampa sono fatte dal ciclo I (95÷165).
Le linee (190÷205) realizzano la visualizzazione del quadrato grigio; che è spostato dalle linee 230 e 245 utilizzando le 4 frecce.
Le linee (250÷280) ristampano la lettera scelta nella posizione del quadrato grigio.
La linea 290 contabilizza il numero di tentativi, che è stampato grazie alle linee 180 e 185.

PER GIOCARE:

Si lancia il programma con RUN. La partita sarà evidentemente tanto più difficile quanto meno si conosce la frase da ricostituire. Si possono spostare le lettere e farle passare nella posizione del quadrato grigio grazie ai 4 tasti di direzione.
Il numero di tentativi è visualizzato in permanenza, a destra del puzzle. Si può fermare il gioco in ogni momento battendo RESTORE.

ESTENSIONI POSSIBILI:

Lo stesso principio può essere applicato ad ogni sorta di puzzles (manipolazioni di numeri, di colori, di suoni eventualmente). Si possono anche prevedere varie frasi, con una scelta effettuata da un generatore aleatorio, ecc.

```
00010 PRINT"PUZZLE di Lettere 094@&"
00015 INPUT "Dammi un Num.INTERO :";F
00020 FOR I=1 TO F:Y=RND(1):NEXT I
00025 Z$=CHR$(127):S$=Z$+Z$:GOSUB 320
00030 INIT 7:PAGE:STORE
00035 FOR I=1 TO 7
00040 CURSORXI+3:CURSORY1:PRINTZ$
00045 CURSORX 3:CURSORY I:PRINTZ$
00050 CURSORX 10:PRINTZ$
00055 CURSORX I+2:CURSORY 8:PRINTZ$
00060 CURSORX2*I+13:CURSORY1:PRINTS$
00065 CURSORX 13:CURSORY I:PRINTS$
00070 CURSORX 27:CURSORY I+1:PRINTS$
00075 CURSORX 13:CURSORY I+6:PRINTS$
00080 CURSORX 27:CURSORY I+7:PRINTS$
00085 CURSORX2*I+11:CURSORY14:PRINTS$
00090 NEXT I:SCREEN:DISPLAY:STORE
00095 FOR I=1 TO 35
00100 X=INT(RND(1)*35+1)
00105 IFMID$(R$,X,1)="-"THEN100
00110 A$=A$+MID$(R$,X,1)
00115 L$=LEFT$(R$,X-1)+"-"
00120 R$=L$+RIGHT$(R$,LEN(R$)-X)
00125 F=ASC(MID$(A$,I,1))
00130 X=I-6*INT(I/6):Y=INT(I/6)
00135 CURSORX X+4:CURSORY Y+2
00140 TX4,0,0:PRINTCHR$(F)
00145 CURSORX 2*X+15:CURSORY 2*Y+2
00150 TX1,3,0:PRINTCHR$(F);CHR$(F)
00155 CURSORX 2*X+15:CURSORY 2*Y+3
00160 PRINTCHR$(F);CHR$(F)
00165 NEXT I:SCREEN:DISPLAY
00170 X=0:Y=0:S=0:C=0
00175 CURSORX 3:CURSORY 17:TX0,0,0
00180 PRINT"N u m e r o  P R O V E ";
00185 PRINT": ";S:TX1,0,0
00190 CURSORX X+4:CURSORY Y+2:PRINTZ$
00195 CURSORX2*X+15:CURSORY2*Y+2
00200 TX2,3,0:PRINTZ$;Z$:CURSORX2*X+15
00205 CURSORY 2*Y+3:PRINTZ$;Z$;TX4,0,0
00210 T$=CHR$(KEY(0))
00215 IF T$="" THEN 300
00220 I=X:J=Y
00230 M=X-(T$=CHR$(61)ANDX<5)
00235 X=M+(T$=CHR$(47)ANDX>0)
00240 N=Y-(T$=CHR$(45)ANDY<5)
00245 Y=N+(T$=CHR$(43)ANDY>0)
00250 C=PEEK(2*(X+4)+80*(Y+2)-32760)
00255 CURSORX I+4:CURSORY J+2
00260 TX4,0,0:PRINTCHR$(C)
00265 CURSORX 2*I+15:CURSORY 2*J+2
00270 TX1,3,0:PRINTCHR$(C);CHR$(C)
00275 CURSORX 2*I+15:CURSORY 2*J+3
00280 FRINT CHR$(C);CHR$(C):TX1,0,0
00285 IF I=X AND J=Y THENSOUND4:GOTO295
00290 S=S+1:SOUND 2:SOUND 4
00295 TX3,0,0:GOTO 175
00300 CURSORXX+4:CURSORYY+2:PRINT" "
00305 CURSORX2*X+15:CURSORY2*Y+2
00310 TX1,3,0:PRINT" ";" ":CURSORY2*Y+3
00315 PRINT" ";" ":TX1.0.0:GOTO 190
00320 R$="HAISPESSOBISOGNODIUNOPIU'"
00330 R$=R$+"PICCOLODITE":A$="":RETURN
```

```
      NSULP          NNSSUULLPP
     OOUOOS          NNSSUULLPP
     DSPISI        OOOOUUOOOOSS
     IBCDIC        OOOOUUOOOOSS
     NOEGPA        DDSSPPIISSII
      O'IIHT       DDSSPPIISSII
                   IIBBCCDDIICC
                   IIBBCCDDIICC
                   NNOOEEGGPPAA
                   NNOOEEGGPPAA
                   OO''IIIIHHTT
                   OO''IIIIHHTT

  N u m e r o  P R O V E :  0
```

Bersagli

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: fortuna.
Difficoltà del gioco: facile.

IL GIOCO:

Consiste semplicemente nello sparare su dei bersagli, cercando di segnare il maggior numero di punti; il numero di bersagli (da 1 a 6) sarà scelto all'inizio della partita, ed il punteggio realizzato sarà visualizzato in permanenza. Se si tocca il bordo di un bersaglio, la partita si ferma. Si marcheranno tanti più punti quanto il numero sul quale si cade sarà grande.

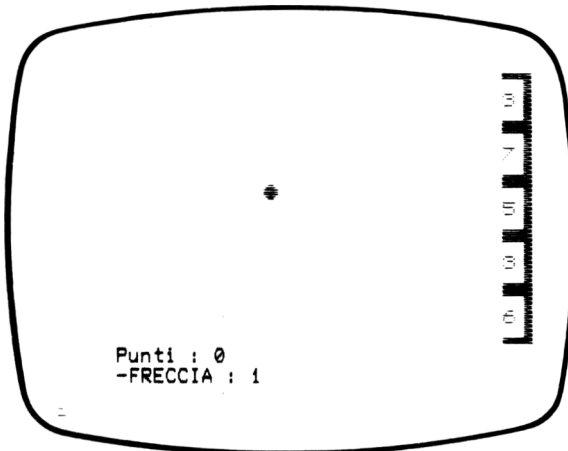
IL PROGRAMMA:

Il numero di bersagli è inserito alla linea 50. Il ciclo I (80÷100) effettua la stampa dei bersagli.
La linea 120 visualizzerà il punteggio in permanenza.
La freccia è visualizzata dalla linea 130.
Il ciclo I (150÷200) realizza la stampa delle cifre contenute nei bersagli, e permette lo spostamento della freccia (linea 160).
La linea 220 guarda se si ha sparato, e la linea 230 guarda se la palla è ancora sulla linea di partenza.
La linea 240 fa avanzare la palla di una casella verso destra, e la linea 250 verifica se ha oppure no raggiunto i bersagli; poi le linee 260 e 270 guardano ciò che hanno colpito.
La linea 280 incrementa il punteggio.
Le linee 290 e 300 si occupano della fine della partita.

PER GIOCARE:

Lanciare il programma con RUN, e battere il numero di bersagli desiderati seguito da RETURN. I bersagli appaiono, e la partita comincia. La vostra freccia, raffigurata dal carattere grafico SETEG 32 è visualizzata in basso a sinistra; potete spostarla con le frecce alto e basso e potete sparare quando volete battendo 0.
Il punteggio sarà modificato in funzione del vostro risultato, e una nuova pallina appare. È possibile fermarsi in ogni istante battendo RESTORE, altrimenti il gioco si ferma dopo un certo numero di spari.

ESTENSIONI POSSIBILI: Si potrà senza difficoltà aumentare il numero di bersagli, modificare le regole di apparizione dei numeri nei bersagli per renderle meno casuali, fare in modo che la pallina stessa sia raffigurata da una cifra che dovrebbe coincidere all'arrivo con quello del bersaglio.

```
00010 PRINT"B E R S A G L I O 095@&"
00020 INPUT"Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 SETEG32,"001C3E7F7F7F3E1C0000
00045 INIT 7
00050 PRINT:INPUT "Num.BERSAGLI :";N
00055 IFN>6THEN50
00060 P=10:J=30:S=0:B=1
00070 INIT 7:GR0,7,0
00080 FOR I=1 TO N:GR0,7,0
00085 CURSORX J:CURSORY 3*I-1:PRINT"Ck"
00090 CURSORX J:CURSORY 3*I:PRINT" j"
00095 CURSORX J:CURSORY 3*I+1:PRINT"pz"
00100 NEXT I:TX0,0,0
00110 K=1:A=16
00120 CURSORX1:CURSORY18:PRINT"Punti";
00125 PRINT" :";S:PRINT"-FRECCIA :";B
00130 CURSORX K:CURSORY A:EG0,7,0
00135 PRINTCHR$(32):TX0,0,0
00140 D=A
00150 FOR I=1 TO N
00160 U=INT(RND(1)*P)
00170 CURSORX J:CURSORY 3*I
00175 PRINTCHR$(U+48)
00180 A$=CHR$(KEY(0))
00190 A=A-(A$=CHR$(45))+(A$=CHR$(43))
00200 NEXT I
00210 CURSORX K:CURSORY D:PRINT" "
00220 IFA$="9"THENB=B+1:GOTO 240
00230 IFK=0THEN130
00240 K=K+1
00250 IFK<JTHEN130
00260 C=PEEK(2*K+80*A-32760):SOUND 1
00270 IFC<48ORC>57THENTX0,2,0:GOTO 320
00280 S=S+C-48:SOUND 0:SOUND 4:SOUND 2
00290 P=P-1
00300 IFP=0THENSOUND4:SOUND4:SOUND4:END
00310 GOTO 110
00320 PRINT"P E R S 0 ! ! ! ":END
```

Gocce

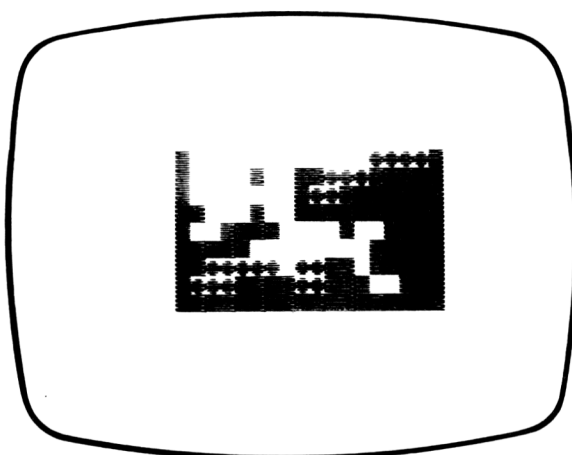
RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: disegno.
Difficoltà del gioco: nessuna.

IL GIOCO: È un vero programma di disegno animato che si può considerare, infatti, come un interludio; un recipiente è disegnato sullo schermo, visto di taglio; e contiene già degli ostacoli. Una goccia d'acqua, simbolizzata da un cerchio grigio, appare sopra il recipiente e cade riempiendolo man mano. La sua traiettoria, quando trova un ostacolo, è interamente determinata dal caso: va a destra o a sinistra, fino a che non ci sia più alcuna possibilità. Allora, un'altra goccia cade, e così via, fino al traboccamento finale.

IL PROGRAMMA: I due cicli chiusi I (60÷130) e J (70÷120) disegnano il recipiente e gli ostacoli.
Le linee 170 e 175 disegnano la goccia.
La linea 190 cancella la goccia precedente e la linea 210 rinvia nel sottoprogramma 310-390 che permette di determinare se c'è o no incontro con un ostacolo.
Se non c'è, si va in 290 e la goccia continua a cadere.
Se c'è, la goccia va a destra o a sinistra (linea 190) secondo il valore definito per K alla linea 150.
Si guarda di nuovo se c'è oppure no l'ostacolo, o se la goccia può continuare a cadere (linee 240, 250, 260).
Secondo il risultato del test, si utilizzerà 230, 270, 290 o 340.

PER GIOCARE: Il RUN fa partire il disegno del recipiente e degli ostacoli, la goccia appare, si mette a cadere e finisce per trovarsi un posto. Un'altra goccia appare e si ricomincia. Il programma non si ferma da solo quando il recipiente trabocca; possiamo fermarlo in qualsiasi momento battendo RESTORE.

ESTENSIONI POSSIBILI: Possiamo, naturalmente, aggiungere il colore ed i suoni, ed anche immaginare un gioco nel quale un piccolo personaggio dovrebbe scappare dalle gocce o dall'annegamento.

```
00010 PRINT" G O C C E  0960&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 SETEG32,"001C3E7F7F7F3E1C0000"
00050 Z$=CHR$(127):INIT 7:PAGE:STORE
00060 FOR I=6TO 12
00070 FOR J=1 TO 16
00080 CURSORX 12:CURSORY I:PRINTZ$
00085 CURSORX 28:CURSORY I:PRINTZ$
00090 CURSORX J+11:CURSORY 12:PRINTZ$
00100 IFI<>6ANDI<>12THENGOSUB 400
00110 IFI<>12ANDRND(1)<.7THENGOSUB 420
00120 NEXT J
00130 NEXT I:SCREEN:DISPLAY
00140 Y=INT(RND(1)+14+6):X=3
00150 K=-2*(RND(1)<.5)-1
00160 I=0
00170 CURSORX Y:CURSORY X:EG0,7,0
00175 PRINTCHR$(32):TX0,0,0
00180 FOR J=0 TO 100:NEXT J
00190 CURSORX Y:CURSORY X:PRINT" "
00200 U=X+1:V=Y
00210 GOSUB 370
00220 IFJ=0THEN290
00230 U=X:V=Y+K
00240 GOSUB 370
00250 IFJ=0THEN310
00260 IFI=0THEN340
00270 CURSORX Y:CURSORY X:EG0,7,0
00275 PRINTCHR$(32)
00280 GOTO 140
00290 X=X+1
00300 GOTO 150
00310 Y=Y+K
00320 I=1
00330 GOTO 170
00340 K=-K
00350 I=1
00360 GOTO 230
00370 C=PEEK(2*V+80*U-32760):SOUND 4
00380 J=-(C<>32):IFU=13THEN440
00390 RETURN
00400 CURSORXJ+12:CURSORYI
00410 PRINTCHR$(127):RETURN
00420 CURSORXINT(RND(1)*14+13):CURSORYI
00430 PRINT"  ":RETURN
00440 CURSORX 16:CURSORY 15:TX0,2,0
00450 PRINT"P i e n o ":SOUND2:END
```

Ostacoli

RIFERIMENTI:

Difficoltà di programmazione: abbastanza difficile.
Categoria del gioco: gioco di riflessi (per un giocatore).
Difficoltà del gioco: media.

IL GIOCO:

Degli ostacoli si spostano sullo schermo per venire incontro ad una piccola macchina simbolizzata da una Y reverse, che dispone della possibilità di spostarsi verso sinistra o destra per evitarli durante il maggior tempo possibile. La distanza percorsa sarà visualizzata in permanenza, ed il gioco termina quando la macchina si scontra con un ostacolo, o col bordo del terreno.

IL PROGRAMMA:

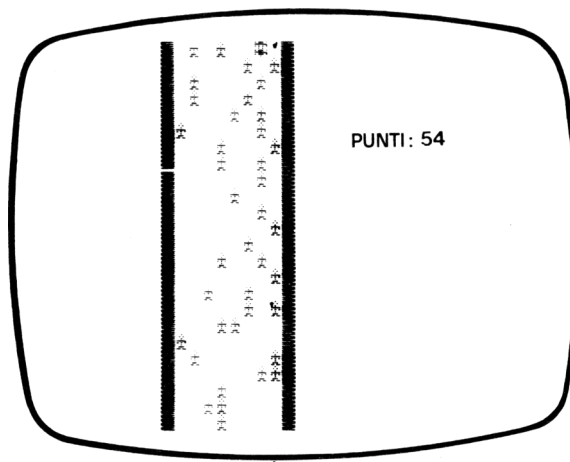
La linea 75 fa avanzare gli ostacoli verso la macchina del giocatore che è stampata grazie alla linea 70.
Le linee (75÷110) fabbricano la strada (le linee 85÷105, anche se identiche, sono tutte e due indispensabili, poiché il valore di RND non sarà, in teoria, lo stesso sulle due linee).
La linea 125 visualizza il punteggio.
Le linee 135 e 140 permetteranno di spostare il veicolo.
Le linee 150 e 160 permettono di controllare se c'è o no un ostacolo e fermano il programma se è necessario.
Il punteggio è incrementato dalla linea 170.

PER GIOCARE:

Si lancia il programma battendo RUN. La strada e gli ostacoli appaiono subito nel basso dello schermo e salgono verso la macchina raffigurata da una Y reverse. La macchina può spostarsi con l'aiuto delle frecce di spostamento orizzontale. Lo scontro con un ostacolo interrompe lo svolgimento del programma e provoca la visualizzazione del punteggio. Si può, però, uscire dal programma in ogni momento utilizzando RESTORE.

ESTENSIONI POSSIBILI:

Sarà abbastanza facile migliorare il disegno della macchina, della strada e degli ostacoli. Potremo anche introdurre il colore, ammettere un certo numero di incidenti, registrare il miglior punteggio (per vari giocatori), eventualmente disegnare varie macchine, ecc.

```
00010 PRINT"O S T A C O L I 097@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FORI=1TOF:Y=RND(1):NEXT I:INIT 7
00040 SETEG32,"5D7F55145D7F5D080000"
00050 SETEG33,"0814083E2A081C360000"
00060 S=0:Y=5:Z$=CHR$(127)
00065 CURSORX Y:CURSORY 1:EG0,7,0
00070 PRINTCHR$(32):CURSORY22:TX0,0,0
00075 PRINTZ$;SPC(8);Z$;SPC(30);" ";
00085 EG0,7,0:IFRND(1)>=.7THEN100
00090 CURSORX INT(RND(1)*8+2):CURSORY22
00095 PRINTCHR$(33):TX0,0,0:GOTO 115
00100 IFRND(1)>=.8THENTX0,0,0:GOTO 115
00105 CURSORX INT(RND(1)*8+2):CURSORY22
00110 PRINTCHR$(33):TX0,0,0
00115 CURSORX20:CURSORY5:PRINT"          ";
00120 PRINT"          ":CURSORX20:CURSORY6
00125 TX0,4,0:PRINT"P U N T I:";S
00130 A$=CHR$(KEY(0))
00135 U=Y-(A$=CHR$(61)ANDY<10)
00140 Y=U+(A$=CHR$(47)ANDY>2)
00150 C=PEEK(2*Y+80*1-32760)
00160 IFC=127ORC=161THENSOUND1:END
00170 S=S+1:GOTO 65
```

Catture nello spazio

RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: riflessi (un solo giocatore).
Difficoltà del gioco: media.

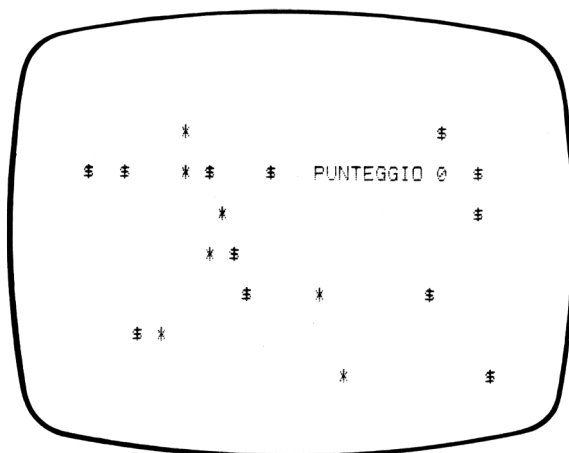
IL GIOCO: La vostra astronave naviga in un pericoloso campo di meteoriti e la vostra missione è di impadronirvi della maggiore quantità possibile di minerale di uranio, un metallo indispensabile che si trova solo in questo luogo pericoloso. Dovete riuscire a manovrare per raccogliere l'uranio, senza farvi urtare da una meteorite.

IL PROGRAMMA: La linea 70 permette di fare apparire le meteoriti.
La linea 90 farà apparire il minerale di uranio.
Si assicura il loro spostamento sullo schermo con la linea 95.
Le linee (110÷130) controllano lo svolgimento del gioco (abbiamo urtato una meteorite? o abbiamo preso un blocco di uranio?).
Lo scontro con una meteorite provoca la fine della partita.
La vostra astronave è stampata dalle linee 145 e 150 ed il suo spostamento è controllato dalle linee 170 e 175.

PER GIOCARE: Si lancia il programma con RUN e le meteoriti cominciano a salire sullo schermo, così come i blocchi di uranio. I tasti di spostamento orizzontale saranno utilizzati per spostare la vostra nave e recuperare l'uranio, ma il minimo urto con una meteorite provocherà la fine della partita e la visualizzazione della quantità di uranio catturata. Si potrà anche fermare in ogni momento battendo RESTORE.

ESTENSIONI POSSIBILI: Si può rendere il gioco sempre più difficile aumentando il numero di meteoriti, o ancora, dando un limite di tempo alla partita. Si può anche prevedere una visualizzazione permanente del punteggio, introdurre una seconda astronave manovrata da una potenza straniera che cercherà anch'essa di impadronirsi dell'uranio...

```
00010 PRINT"Cattura SPAZIALE 0980&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FORI=1TOF:Y=RND(1):NEXT I:INIT 7
00040 SETEG32,"00000808367F1C080000"
00050 X=15:Z=0
00060 S=INT(RND(1)*38)+1:TX1,0,0
00070 CURSORXS:CURSORY21:PRINT"$";
00080 S=INT(RND(1)*38)+1:TX2,0,0
00085 IFRND(1)>=.7THEN100
00090 CURSORXS:CURSORY21:PRINT"*";
00095 PRINT:PRINT
00100 FOR I=1 TO 10:NEXT I
00110 C=PEEK(2*X+80*11-32760)
00120 IFC=42THENSOUND7:SOUND1:GOTO190
00130 IFC=6THENZ=Z+1:SOUND1:SOUND7
00135 TX4.0.0:CURSORXX-1:CURSORY10:
00140 PRINT"   ":TX3.0.0:CURSORX X-1
00145 CURSORY 12:PRINT" "::EG4.7.0
00150 PRINTCHR$(32)::TX3.0.0:PRINT" "
00160 A$=CHR$(KEY(0))
00170 W=X-(A$=CHR$(61)ANDX<39)
00175 X=W+(A$=CHR$(47)ANDX>1)
00180 GOTO 60
00190 CURSORX20:CURSORY9:TX0.4.0
00200 PRINT"PUNTEGGIO":Z:SOUND 4:END
```

PUNTEGGIO 0

Anomalia

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: solitario.
Difficoltà del gioco: media.

IL GIOCO:

Si basa essenzialmente su una buona capacità di osservazione. Due griglie di 25 caselle ciascuna sono disegnate sullo schermo. Dei simboli identici sono visualizzati in posti differenti delle due griglie, ma uno dei simboli è stato modificato sulla griglia di destra. Il gioco consiste nel trovare quale, inserendo le sue coordinate nell'ordine linea-colonna.

IL PROGRAMMA:

Il ciclo I (50÷95) stampa le cifre di riferimento delle griglie.
Il ciclo principale I (100÷160) stampa i simboli delle due griglie, in un ordine differente.
La linea 110 sceglie il carattere da stampare.
Il sottoprogramma (300÷350) si occupa di fare la stampa in una casella libera (lo si richiama due volte per le due griglie).
Le linee 170 e 180 cambiano, in maniera aleatoria, uno dei caratteri.
Le coordinate del punto tentato sono inserite alle linee 190 e 200, ed il controllo è effettuato dalle linee 210 e 220.
Le linee 230÷260 si occupano della visualizzazione del risultato.

PER GIOCARE:

Dopo aver lanciato il programma con l'istruzione RUN, si ottiene la visualizzazione di due griglie. Si dovrà allora cercare, per paragoni, qual è il carattere che è stato modificato (sulla griglia di destra).
Si inserisce allora il suo numero di linea, seguito da RETURN, ed il suo numero di colonna, seguito da RETURN. Il programma risponderà allora BRAVO o PERSO.

ESTENSIONI POSSIBILI:

Se si dispone del colore, sarà possibile utilizzare dei simboli grafici di differenti colori, il che potrà semplificare la ricerca. Si può anche immaginare di utilizzare dei disegni più sofisticati, migliorare il controllo dei risultati, ammettere la buona risposta sulla griglia di destra o di sinistra, introdurre un controllo di tempo, ecc.

```
00010 PRINT" A N O M A L I A  099@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE:T=5:STORE:TX1,4,0
00050 FOR I=1 TO 5
00060 CURSORX 2*I+1:CURSORY 1:PRINTI
00070 CURSORX 1:CURSORY 2*I+1:PRINTI
00080 CURSORX 2*I+19:CURSORY 1:PRINTI
00090 CURSORX 18:CURSORY 2*I+1:PRINTI
00095 NEXT I:SCREEN:DISPLAY:TX0,0,0
00100 FOR I=1 TO 22
00110 D=INT(26*RND(1)+1)
00120 L=2:S=T
00130 GOSUB 300
00140 L=11
00150 GOSUB 300
00160 NEXT I
00170 S=0
00180 GOSUB 300
00190 CURSORX1:CURSORY15:INPUT"LINEA";I
00200 INPUT "COLONNA";T
00210 U=2*I+1:V=2*T+20
00220 C=PEEK(2*V+80*U-32760):PRINTC
00230 IFD+64=CTHENPRINT"BRAV0":GOTO 250
00240 PRINT:PRINT"PERSO":SOUND 0
00250 PRINT"La LETTERA era :";
00260 PRINTCHR$(D+64):SOUND 2:END
00300 U=2*INT(T*RND(1))+3
00310 V=2*INT(S*RND(1)+L)
00320 C=PEEK(2*V+80*U-32760)
00330 IFC<>32ANDS<>0THEN300
00340 CURSORX V:CURSORY U
00350 PRINTCHR$(D+64):RETURN
```

```
   1 2 3 4 5          1 2 3 4 5
 1 X B R T K      1   R V B P
 2 A U A J V      2   A Z R S D
 3 P U     Z      3   J M A   A
 4 V A B V        4   S U A   V
 5 D R M A S      5   V U T B K


LINEA? 2
COLONNA? 3
 82

PERSO
La LETTERA era :S
```

Tanti in uno

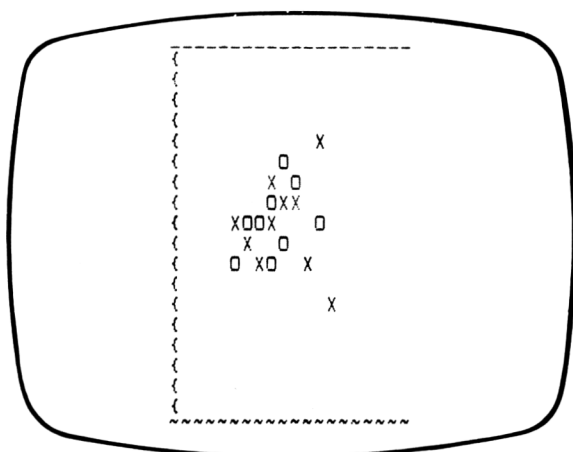
RIFERIMENTI: Difficoltà di programmazione: difficile.
Categoria del gioco: due giocatori.
Difficoltà del gioco: variabile, secondo l'utilizzo.

IL GIOCO: Molti giochi sono in realtà possibili con questo programma che si comporterà come una matita elettronica, che vi permetterà di scrivere dei simboli O o X, in un qualunque posto del foglio-schermo. Potremo, per esempio, utilizzarlo per un gioco da ragazzi (il primo a realizzare un allineamento di 5 pedine identiche ha vinto) o per un GO (si realizzano gli accerchiamenti delle pedine avversarie) o un altro gioco, secondo la vostra immaginazione.

IL PROGRAMMA: Il campo di gioco è disegnato dai cicli I (40÷50) e J (60÷80).
Il cambio del giocatore è effettuato dalla linea 100. Il giocatore che gioca vede il simbolo che ha scelto (O o X) spostarsi al limite del campo (linee 120 e 130).
Le linee (140÷190) permettono, ai due giocatori, di spostare il simbolo sul terreno.
Le linee (220÷230) controllano se lo spostamento è possibile (casella libera e non oltrepasso dei limiti).
La linea 290 permette di scrivere su una casella libera.

PER GIOCARE: RUN provoca l'apparizione dei limiti del campo e del numero del giocatore, così come i simboli corrispondenti (O o X) che si dovranno spostare sullo schermo con l'aiuto dei 4 tasti di direzione. Non è possibile andare su una casella già occupata e bisognerà battere 0 per iscrivere il proprio simbolo in una casella disponibile.
Fatto questo, il simbolo del secondo giocatore appare nella colonna di sinistra, circa alla metà, ed il giocatore procederà nella stessa maniera del suo avversario. Per uscire dal programma, basterà battere RESTORE.

ESTENSIONI POSSIBILI: Si potrà utilizzare l'intero schermo come terreno di gioco e realizzare dei controlli in funzione dei differenti giochi che si desiderano fare: controllo di allineamento di 5 simboli per il gioco di cui parlavamo prima, controllo del numero di pedine catturate per il GO, ecc.

```
00010 PRINT"TANTI in UNO 100@&"
00020 S$=CHR$(123):Z$=CHR$(95)
00030 INIT 7:T$=CHR$(126):PAGE:STORE
00040 FOR I=1 TO 20
00050 CURSORX10:CURSORYI:PRINTS$;:NEXTI
00060 FOR J=10 TO29
00070 CURSORXJ:CURSORY 1:PRINTZ$
00080 CURSORXJ:CURSORY20:PRINTT$;:NEXTJ
00090 J=0:SCREEN:DISPLAY
00100 J=ABS(J-1)
00110 X=10:Y=30
00120 CURSORX Y:CURSORY X:TX0,4,0
00130 PRINTCHR$(79-9*(J=1))
00140 A$=CHR$(KEY(0)):IFA$=""THEN140
00150 A=X:B=Y
00160 M=X-(A$=CHR$(45)ANDX<19)
00170 X=M+(A$=CHR$(43)ANDX>2)
00180 N=Y-(A$=CHR$(61)ANDY<29)
00190 Y=N+(A$=CHR$(47)ANDY>11)
00200 CURSORXB:CURSORYA:TX0,0,0
00210 PRINT" "
00220 C=PEEK(2*Y+80*X-32760)
00230 IFC<>79ANDC<>88THEN250
00240 X=A:Y=B
00250 A$=CHR$(KEY(0))
00260 IFA$=""THEN250
00270 IFA$<>"9" ORY=1THEN120
00280 CURSORX Y:CURSORY X
00290 PRINTCHR$(79-9*(J=1))
00300 IFJ=0THENSOUND 0:GOTO 320
00310 IFJ<>0THENSOUND 4:SOUND 6
00320 GOTO 100
```

Salti di Cavallo

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessione.
Difficoltà del gioco: difficile.

IL GIOCO:

Si presenta, qui, come un gioco totalmente di fortuna, ma proviene, in realtà, da un problema assai difficile da risolvere: come spostare un cavallo su una scacchiera in modo che passi una volta, e una sola, su ognuna delle 64 caselle della scacchiera. Il cavallo si sposta secondo le regole comunemente ammesse negli scacchi, e la sua posizione di partenza è definita dal giocatore; i suoi spostamenti, invece sono effettuati a caso. Riuscirà a percorrere le 64 caselle?

IL PROGRAMMA:

Le linee 50 e 60 definiscono i diversi spostamenti possibili, in due variabili alfanumeriche A$ e B$; il ciclo chiuso I (70÷110) disegna il campo di gioco.
Si inseriscono le coordinate della casella di partenza (linea poi colonna) alle linee 120 e 130.
Le linee (150÷170) stampano il numero della mossa.
La linea 180 permette la pesca a caso di uno spostamento del cavallo.
Il ciclo Z (190-250) cerca se lo spostamento è possibile; se non è il caso, si cercano tutte le altre (sempre nello stesso ordine).
Le linee 200 e 210 effettuano lo spostamento.
La linea 220 controlla che non si esca dalla scacchiera.
Le linee 230 e 240 inviano alla linea 280 se il colpo è possibile.
La linea 270 gestisce la fine della partita (cavallo bloccato) e visualizza il punteggio.
La linea 280 realizza la mossa e incrementa il punteggio.

PER GIOCARE:

Si lancia il programma con l'ordine RUN. Il campo di gioco appare; si sceglie allora la posizione di partenza battendo prima un numero di linea (tra 1 e 8), seguito da RETURN, poi il suo numero di colonna, seguito da RETURN. Il gioco comincia immediatamente e si svolge in modo interamente automatico fino a che il cavallo sia bloccato. Si può, naturalmente, fermarsi in ogni momento con RESTORE.

ESTENSIONI POSSIBILI: Si possono immaginare numerose varianti su questo tema, cominciando col far variare le dimensioni della scacchiera, si può anche migliorare considerevolmente il grafismo, utilizzare il colore. Ma, la cosa più interessante, è di riuscire a definire un algoritmo che dia la o le soluzioni del problema iniziale...

```
00010 PRINT"SALTI di CAVALLO 101@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 INIT 7:PAGE:STORE
00050 S=1:T=0:A$="56653223"
00060 B$="653223"+A$:A$=A$+A$
00070 FOR I=1 TO 8
00080 CURSORX 3*I+1:CURSORY 0:PRINTI
00090 CURSORX 1:CURSORY 2*I:PRINTI;
00100 PRINT"  .  .  .  .  .  .  .  ."
00110 NEXTI:CURSORX1:CURSORY18
00115 SCREEN:DISPLAY
00120 INPUT"LINEA di Partenza:";I:PRINT
00130 INPUT"COLONNA di Partenza:";J
00140 IFS>9THENS=0:T=T+1
00150 CURSORX 3*J+2:CURSORY 2*I
00160 PRINTRIGHT$(STR$(S),1)
00170 IFT<>0THENCURSORX 3*J+1:GOSUB 290
00180 Z=INT(RND(1)*8+1)
00190 FOR Z=Z TO Z+8
00200 A=I+VAL(MID$(A$,Z,1))-4
00210 B=J+VAL(MID$(B$,Z,1))-4
00220 IFA<1ORB<1ORA>8ORB>8THEN250
00230 C=PEEK(2*(3*B+2)+160*A-32760)
00240 IFC=46THENZ=Z+8:NEXT Z:GOTO 280
00250 NEXT Z
00260 CURSORX 26:CURSORY 19
00270 PRINT"TOTALE :";T*10+S:SOUND2:END
00280 I=A:J=B:S=S+1:SOUND 4:GOTO 140
00290 CURSORY2*I:PRINTRIGHT$(STR$(T),1)
00300 RETURN
```

```
  1  .  .  7 30  .  .  .  4

  2 10 29  .  1  6  3  .  .

  3  .  . 11  8 25  .  5  .

  4 28  9 26  .  2  .  .  .

  5 15 12  . 24  .  . 19  .

  6  . 27 14 17 20 23  .  .

  7 13 16 21  .  . 18  .  .

  8  .  .  .  . 22  .  .  .

 LINEA di Partenza:? 2
                              TOTALE : 30
 COLONNA di Partenza:? 4
```

Muro di mattoni

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: riflessi.
Difficoltà del gioco: difficile.

IL GIOCO:

Se c'è un gioco che si può presentare come un grande classico dei videogiochi per microcalcolatore, è proprio il muro di mattoni (in inglese BREAKOUT). È offerto, qui, in una versione semplificata, ma offre lo stesso delle buone prestazioni: visualizzazione del punteggio a fine partita, rimbalzi aleatori, ecc.
La regola è molto elementare: disponete di una racchetta che può essere spostata orizzontalmente per rinviare una pallina (materializzata da un punto) che demolirà i mattoni disegnati nell'alto dello schermo. Il vostro problema è di demolirne il più grande numero possibile, senza mancare la pallina, il che provocherebbe la fine della partita e la visualizzazione del punteggio.

IL PROGRAMMA:

Il ciclo I (60÷80) disegna il campo di gioco ed i mattoni sono messi in ordine dai cicli chiusi I (110÷140) e J (120÷140).
La linea 160 disegna la racchetta la cui posizione è controllata dalle linee 180 e 190.
Le linee (220÷280) guardano se si ha colpito il muro, i mattoni o la racchetta.
I rimbalzi sono gestiti di conseguenza in maniera eventualmente aleatoria, dalle linee 300 e 350.
La linea 210 visualizza il punteggio in permanenza.

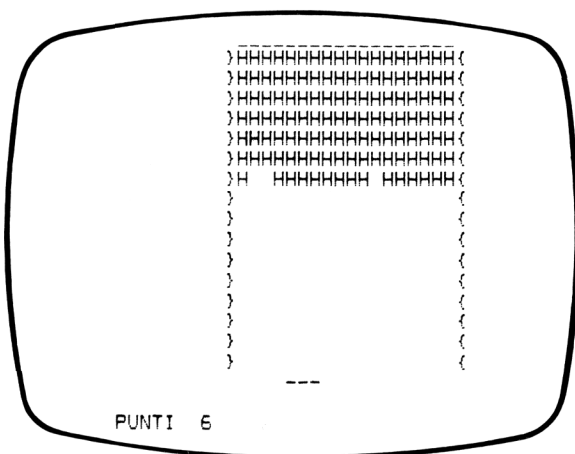
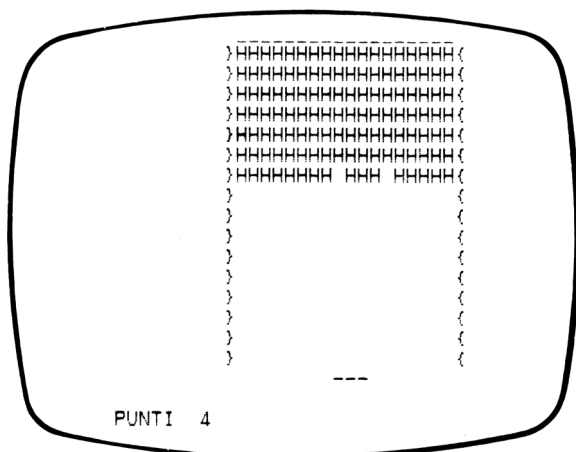
PER GIOCARE:

Il programma sarà lanciato dall'istruzione RUN, il che provoca l'apparizione del campo di gioco, poi, molto rapidamente, l'invio della pallina. Il giocatore dovrà allora cercare di spostare la racchetta con l'aiuto dei tasti di spostamento orizzontale (frecce destra e sinistra). Se, in un momento qualunque, la pallina non tocca la racchetta, il punteggio apparirà e la partita si termina. Si può anche terminare in ogni istante con RESTORE.

ESTENSIONI POSSIBILI:

Si può immaginare di ingrandire il campo di gioco, aumentare il numero di mattoni, prevedere varie palline, disegnare ogni fila di mattoni di un colore diverso, introdurre degli effetti sonori.

Si può anche, ispirandosi ai giochi dei bar, introdurre dei piccoli operai che verranno a ricostruire il vostro muro di mattoni, man mano che voi lo demolite.

```
00010 PRINT"MURO di MATTONI 102@&"
00020 INPUT "Dammi un Num.INTERO :";F
00030 FOR I=1 TO F:Y=RND(1):NEXT I
00040 S$=CHR$(123):Z$=CHR$(125)
00050 T$=CHR$(95):INIT 7:PAGE:STORE
00060 FOR I=2 TO 17
00070 CURSORX 10:CURSORY I:PRINTZ$;
00080 CURSORX29:CURSORYI:PRINTS$;:NEXTI
00090 FOR J=11 TO 28:CURSORXJ:CURSORY 1
00100 PRINTT$;:NEXT J:TX0,4,0
00110 FOR I=2 TO 8
00120 FOR J=11 TO 28
00130 CURSORX J:CURSORY I:PRINT"H"
00135 CURSORY 18:PRINTCHR$(127)
00140 NEXT J:NEXT I:SCREEN:DISPLAY
00150 H=1:K=-.5:J=19:I=16:Y=J:TX0,0,0
00160 CURSORXY-2:CURSORY18:PRINT" --- "
00170 A$=CHR$(KEY(0))
00180 U=Y-(A$=CHR$(61) AND Y<28)
00190 Y=U+(A$=CHR$(47) AND Y>11)
00200 IFT=1 THEN CURSORX J:GOSUB 400
00210 CURSORY 20:PRINT"PUNTI ";S
00220 IFI>18 THEN SOUND 2:END
00230 I=I+H:J=J+K:T=1
00240 C=PEEK(2*J+80*I-32760):CURSORX J
00250 IFC=123 ORC=125 THEN 320
00260 IFC=95 THEN 310
00270 IFC=45 THEN SOUND 4:GOTO 300
00280 IFC=72 THEN SOUND 7:GOTO 330
00290 CURSORY I:PRINT"o":GOTO 160
00300 K=RND(1)-RND(1)
00310 H=-H
00320 K=-K:T=0:GOTO 160
00330 S=S+10-I
00340 H=-H
00350 K=-(K-(2*(RND(1)<.5)+1)*RND(1)/3)
00360 GOTO 160
00400 CURSORY I:PRINTCHR$(32):RETURN
```

Othello

RIFERIMENTI:

Difficoltà di programmazione: difficile.
Categoria del gioco: gioco di riflessione (due giocatori).
Difficoltà del gioco: difficile.

IL GIOCO:

Diventato un grande classico dei giochi su microcalcolatore, il gioco OTHELLO o REVERSI ci guadagna ad essere gestito su calcolatore, con visualizzazione, anche se, come qui, è praticato da due giocatori; in effetti, il gioco consiste essenzialmente nel rigirare le pedine dell'avversario, e questo può rivelarsi, alla lunga molto fastidioso, soprattutto se siete un giocatore assiduo. Il principio è che il giocatore, possedente le X, deve poter giocare, riuscire a chiudere una o più pedine dell'avversario, in una o più direzioni; se ci riesce, le O saranno allora rigirate per diventare delle X (il rigiro potrà essere fatto orizzontalmente, verticalmente, o seguendo le diagonali). Il giocatore che ha le O procederà nella stessa maniera, e questo fino a che la scacchiera sarà riempita o che nessuno possa più giocare (se un giocatore non può girare le pedine, deve cedere il turno).

IL PROGRAMMA:

I cicli chiusi P (50÷130) e Q (65÷130) disegnano il campo di gioco e le linee 110 e 120 visualizzano una X o una O, secondo il giocatore che comincia.
Il ciclo P (190-300) controlla la validità della mossa (la casella è libera? : linee 280 e 290), utilizzando una parte del programma che si trova alle linee 350-410 (abbiamo preso uno o vari nemici alla linea 350 con eventuale rigiro, autorizzato dalla linea 370: si rifà un'esplorazione in tutte le direzioni, se il bloccaggio si è rivelato possibile).
La linea 190 definisce le 8 direzioni di esplorazione.
Le linee 230 e 240 calcolano gli spostamenti da effettuare per esplorare queste 8 direzioni.
La linea 220 realizza gli spostamenti.
La linea 320 garantisce il cambio del giocatore, se c'è stato rigiro (controllo alla linea 310).

PER GIOCARE:

Dopo RUN, ogni giocatore inserirà a turno le coordinate (linea, RETURN, poi colonna, RETURN) della casella dove desidera giocare; se il colpo è accettato, le pedine sono rigirate e si cambia giocatore; se una mossa non è possibi-

le, si inserisce 0 come prima coordinata per il cambio del giocatore.

ESTENSIONI POSSIBILI: Prima di tutto, naturalmente, migliorare la visualizzazione con dei colori, fare una griglia più spaziosa. Si potrà anche immaginare un gioco macchina per il Videopac.
Nota: Dopo aver lanciato il programma con RUN, conviene inserire le dimensioni della griglia (4, RETURN o 6, 8...).

```
00010 PRINT" O T H E L L O  103@&"
00020 INPUT "Dimensione GRIGLIA :":A
00030 IFA/2<>INT(A/2)ORA>10THEN20
00040 INIT 7:PAGE:STORE
00050 FOR P=1 TO A
00055 CURSORX 2*P-1:CURSORY 0:TX0.4.0
00060 PRINTRIGHT$(STR$(P).2)
00065 FOR Q=1 TO A
00070 CURSORX 2*A+2:CURSORY 2*P:TX0,4.0
00075 PRINTRIGHT$(STR$(P).2)
00080 CURSORX 2*Q:CURSORY 2*P:TX0.0.0
00085 PRINT"."
00090 IFQ=A/2+1ANDP=A/2THENGOTO 100
00095 GOTO 110
00100 CURSORX 2*Q-2:CURSORY 2*P:TX0.0,0
00105 PRINT"O ";:TX0,0.0:PRINT"X"
00110 IFQ=A/2+1ANDP=A/2+1THEN120
00115 GOTO 130
00120 CURSORX 2*Q-2:CURSORY 2*P:TX0.0.0
00125 PRINT"X ";:TX0,0,0:PRINT"O"
00130 NEXT Q:NEXT P:SCREEN:DISPLAY
00135 K=1
00140 CURSORX25:CURSORY6:PRINT"Giocat":
00145 PRINT"ore  O":IFK<>1THEN155
00150 CURSORX 35:CURSORY 6:PRINT" X"
00155 CURSORX33:CURSORY8:PRINT"        "
00160 CURSORX35:CURSORY10:PRINT"      "
00165 CURSORX25:CURSORY 8
00170 INPUT "LINEA :";U
00175 IFU=0THEN340
00180 CURSORX25:CURSORY10:
00185 INPUT "COLONNA :";V
00190 H=1:GOSUB 1000
00200 FOR P=1 TO 8
00210 S=0
00220 W=0
00230 T=2*U:R=2*V
00240 F=(P>3ANDP<7)-(P<3ORP>7)
00250 G=(P>1ANDP<5)-(P>=6)
00260 Q=46-42*(S=1)-33*(S=2)
00270 IFS=0THEN290
00280 CURSORXR:CURSORYT:PRINTCHR$(Q)
00290 T=T+2*F:R=R+2*G
00300 C=PEEK(2*R+80*T-32760)
00310 IFC=79ORC=88THEN360
00320 NEXT P
00330 IFH<>0THEN140
00340 K=ABS(K-1)
00350 GOTO 140
00360 IF(C=79ANDK=1)THENW=1:GOTO 370
00365 IF(C=88ANDK=0)THENW=1
00370 IFS<>0THEN430
00375 IF(C=79ANDK=1)ANDW=1THEN290
00380 IF(C=88ANDK=0)ANDW=1THEN290
00390 IFW=0ORS<>0THEN320
00400 S=1-(K=0)
00410 H=-H*(S=0)
00420 GOTO 220
00430 CURSORX R:CURSORY T:PRINTCHR$(Q)
00440 GOTO 375
01000 REMscreenCOPY for G7420-TURBO/VCG
01010 PAGE:CURSORX1:CURSORY0:PRINT".":
01020 PRINTTAB(39):"."
01030 CURSORX1:CURSORY22:LPRINT".":
01040 PRINTTAB(39):"."
01050 FOR I=0 TO 22
01060 FOR J=1 TO 39
01070 C=PEEK(2*J+80*I-32760)
01080 LPRINTTAB(J):CHR$(C):
01090 NEXT J:LPRINT:NEXT I
01100 RETURN
```

```
. . . . . . . . . . . . 1
. . . . . . . . . . . . 2
. . . . . . . . . . . . 3 Giocatore  X
. . . . . . . . . . . . 4 LINEA :? 7
. . . . . O X . . . . . 5 COLONNA :? 3
. . . . . X O . . . . . 6
. . . . . . . . . . . . 7
. . . . . . . . . . . . 8
. . . . . . . . . . . . 9
. . . . . . . . . . . . 10
```

Imparare divertendosi è l'obiettivo di questo libro. Sul filo dei 102 programmi di giochi, vi guiderà alla scoperta del C7420. I programmi sono classificati, a seconda del grado di difficoltà, in cinque livelli e ciascuno fa appello a nuove conoscenze e ad una sempre più approfondita conoscenza del Basic. Ogni livello comincia con una presentazione concisa delle nuove istruzioni utilizzate. Tutti i giochi sono ampiamente descritti e i programmi sono commentati. Per ogni versione è dato un esempio di esecuzione.

★

L. 21.000

I.S.B.N. 88.7688.**102**.6

102 PROGRAMMI PER PHILIPS C7420 VIDEOPAC+
J. DECONCHAT
V. C. GRANDIS
E.P.S.I.